This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

THIS PAGE BLANK (USPTO)

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 690 376 B1

(12) EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention of the grant of the patent:
10.11.1999 Bulletin 1999/45

(51) Int Cl.[6]: **G06F 9/46**, G06T 7/20

(21) Application number: **95114932.7**

(22) Date of filing: **17.02.1989**

(54) Digital signal processing method

Digitalsignalverarbeitungsverfahren

Méthode de traiter des signaux numériques

(84) Designated Contracting States:
DE FR GB IT

(30) Priority: **19.02.1988 JP 3792188**
18.03.1988 JP 6369588
26.11.1988 JP 29872288
26.11.1988 JP 29872388

(43) Date of publication of application:
03.01.1996 Bulletin 1996/01

(62) Document number(s) of the earlier application(s) in accordance with Art. 76 EPC:
89102717.9 / 0 329 151

(73) Proprietor: **MITSUBISHI DENKI KABUSHIKI KAISHA**
Tokyo 100 (JP)

(72) Inventors:
- **Murakami, Tokumichi, c/o Mitsubishi Denki K.K.**
  Kamakura-shi, Kanagawa (JP)
- **Kamizawa, Koh, c/o Mitsubishi Denki K.K.**
  Kamakura-shi, Kanagawa (JP)
- **Kinjo, Naoto, c/o Mitsubishi Denki K.K.**
  Kamakura-shi, Kanagawa (JP)

(74) Representative: **Eisenführ, Speiser & Partner**
Martinistrasse 24
28195 Bremen (DE)

(56) References cited:
EP-A- 0 205 091
US-A- 4 661 849
US-A- 4 667 233

- **SYSTEMS, COMPUTERS, CONTROLS, vol.12, no.4, July 1981, WASHINGTON US pages 37 - 45 ARIKI 'Moving picture analysis based on a hierarchical model approach'**

Note: Within nine months from the publication of the mention of the grant of the European patent, any person may give notice to the European Patent Office of opposition to the European patent granted. Notice of opposition shall be filed in a written reasoned statement. It shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

EP 0 690 376 B1

Printed by Jouve, 75001 PARIS (FR)

Description

BACKGROUND OF THE INVENTION

Field of the Invention

[0001] The present invention relates to a digital signal processing method which performs computational processes for digital signals.

Description of the Prior Art

[0002] Fig. 1 shows the multiprocessor system described in article entitled "A Real Time Video Signal Processor Suitable for Motion Picture Coding Applications", IEEE, GLOBCOM '87, p. 453. In Fig. 1, input data 1 is received by a data transfer controller 3, and thereafter data 4 are transferred selectively to digital signal processors 2, i.e. DSP-1 through DSP-N, in block-1. After being processed by the respective DSPs in block-1, resultant data 5 is transferred to block-2 and processed by respective DSPs for the next processing step.

[0003] Fig. 2(a) shows divided memory areas of the DSPs. For the simplicity of explanation, shown here is an example of parallel processing using three DSPs 2, to which process areas A, B and C are assigned evenly.

[0004] In the inter-frame image coding system and the like, it is a general convention to employ the conditional pixel supplementary process in which only portions having at least a certain difference between the input frame and previous frame are coded and previous frame data is used for the remaining portions. Accordingly, the volume of computation needed for the process differs depending on the valid pixel rate even though the number of pixels in the process area is constant. The volume of computation or computation time needed is proportional to the valid pixel rate.

[0005] In the inter-frame image coding system or the like, assuming that the number of valid pixels is shared by all DSPs to have a distribution EA, EB and EC as shown in Fig. 2(b), the computation time needed for one block of parallel DSP configuration is determined from the process time of the DSP which works for the area B with the largest volume of process M, and the remaining DSPs which have finished the areas A and C earlier have idle time.

[0006] The conventional digital signal processing apparatus arranged as described above has its overall process time determined from the longest process time among DSPs when the density of information, such as the valid pixel rate, within a frame is uneven and the distribution of information varies with time, resulting in a degraded process efficiency per DSP unit.

[0007] Fig. 3 is a diagram showing, as an example, the arrangement of other digital signal processing apparatus disclosed in an article entitled "Realtime Video Signal Processor Module", in the proceeding of ICASSP '87, pp. 1961 - 1964, April 1987, Dallas, U.S.A. In the figure, indicated by 1 is an input terminal, 4 is an input bus for distributing input data on the input terminal 1, 28a is a feedback bus for distributing the result of previous process, and 20 are signal processing modules each including an input storage 21, a processing unit 22, an output storage 23 and a timing control unit 24. Indicated by 25 are wired-OR circuits through which feedback data on output ports 30 are placed on the feedback bus 28a, 26 are wired-OR circuits through which output data on output ports 29 are delivered to the output terminal 5 over the output bus 5a, 27 are input ports for the input data to the signal processing module 20, and 28 are input ports for the feedback data to the signal processing module 20.

[0008] Fig. 4 is a block diagram showing in more detail one of the signal processing module in Fig. 3. In the figure, indicated by 221 is an address generator (AGU A), 211 is an input dual memory (MEM A) which receives data on the input port 27 over the input bus 4, 212 is an input dual memory (MEM B) which receives data on the feedback bus 28a by way of the input port 28, 222 is an address generator (AGU B), 223 is an X-bus, 224 is a Y-bus, and 225 is a pipeline arithmetic unit (PAU) having its input terminal EX1 connected to the X-bus 223 and another input terminal EX2 connected to the Y-bus 224. Indicated by 226 is a data memory [MEM P(Q)] having its output connected to the X-bus 223, 227 is an address generator [AGU P(Q)] having its output connected to the Y-bus 224 and data memory 226, 228 is a mode register (MDR) having its output connected to the X-bus 223 and Y-bus 224, and 241 is a Z-bus connected to the inputs of the address generators 221, 222 and 227, pipeline arithmetic unit 225 and data memory 226. Indicated by 242 is a sequencer (SEQ), 243 is an instruction memory (IRAM) connected to the output of the sequencer 242, and 245 is a decoder (DEC) connected to the output of the instruction memory 243, with the output of the decoder 245 being connected to the Z-bus 241 and output bus 231. The output bus 231 is connected to the input of the mode register 228 and the Z-bus 241. Indicated by 232 is an FIFO memory (MEM C) connected to the output bus 231, 233 is an FIFO (MEM D) connected to the output bus 231, 29 is an output port of the FIFO memory 232, and 30 is an output port of the FIFO memory 233.

[0009] Fig. 5 is a diagram showing, as an example, the algorithm of a typical high-efficiency coder for a moving image. In the figure, indicated by 250 is an input terminal for the input video signal, 251 is an input frame buffer having at least a 1-frame capacity and having the simultaneous read-write ability, 252 is an inter-frame subtracter for evaluating

the difference, 253 is a block identifier, 254 is a coder, 255 is a coding parameter produced by the coder 254, 256 is a variable-length coder, 257 is a video multiplexer, 258 is a transmission buffer memory, and 259 is an output terminal for the coded data. Connected in cascade between the input terminal 250 and output terminal 259 are the above-mentioned functional blocks 251 - 254 and 256 - 258. Further indicated by 260 is a local decoder which receives the coding parameter 255, 261 is an inter frame adder, 262 is an in-loop filter, 263 is a coding frame memory, 264 is previous coded frame data, 265 is a motion compensator, 266 is current frame data fed from the input frame buffer 251 to the motion compensator 265, 267 is motion vector data, 268 is compensated previous frame data fed from the motion compensator 265 to the inter-frame subtracter 252 and inter-frame adder 261, 269 is a feedback signal, and 270 is a coding controller which provides coding control information for the video multiplexer 257, a feed-forward signal to the input frame buffer 251, a block identification control signal 273 to the block identifier 253, and a coding control signal 274 to the variable-length coder 256.

[0010] Next, the operation of the conventional digital signal processing apparatus will be described in connection with Fig. 3. This apparatus is intended for moving image processing and is based on the division parallel processing system in which a frame is divided into small frames and a signal processing module 20 is assigned to each of the divided frame areas.

[0011] Initially, each signal processing module 20 operates on the autonomous basis by expending one video frame time to fetch a divided frame area assigned to it among the input data transferred frame-wise in raster scanning over the input bus 4 and store the data in the input storage 21. At the same time, if the process result of the previous frame is needed for the current process, it operates by expending one video frame time to fetch data of the assigned area of the frame in the feedback data from the input port 28 over the feedback bus 28a and stores the data in the input storage 21.

[0012] Upon expiration of one video frame time, the processing unit 22 performs the prescribed signal processing for the input data and feedback data stored in the input storage 21, and stores the result temporarily in the output storage 23. The feedback data led out of the output storage 23 through the output port 30 is timed for synchronization with other signal processing modules 20 and, by being merged into all feedback data by the wired-OR circuit 25, placed on the feedback bus 28a. Similarly, the output data led out of the output storage 23 through the output port 29 is timed for synchronization with other signal processing modules 20 and, by being merged into all output data by the wired-OR circuit 26, delivered to the output terminal 5 over the output bus 5a.

[0013] Divided frame areas processed individually by the signal processing modules 20 are combined back to a video frame. Therefore, parallel processing of area divided type is realized. For reason as described above, it is necessary for all signal processing modules 20 to have their process commencement in complete synchronism with one another. On this account, the timing control unit 24 provides all sections of system with the timing of data input/output and process commencement in synchronism with the video frame timing which is the synchronization reference point.

[0014] Next, the operation of one signal processing module 20 will be briefed in connection with Fig. 4. Among a video frame entered frame-wise through the input port 27 in synchronism with the video frame sync signal, data of the assigned area is stored in the input dual memory 211. At the same time, among the coded previous frame data entered through the input port 28, the portion of the assigned area and its peripheral data are stored in the input dual memory 212.

[0015] The input dual memories 211 and 212 is made up of a two-sided memory device in the same structure on both sides and it operates such that while one side is written data, the other side is connected to the X-bus 223 and Y-bus 224 for reading for the coding process by the pipeline arithmetic unit 225. The read/write sides of the input dual memories 211 and 212 are switched by the above-mentioned video frame sync signal so that input data of assigned areas on the input ports 27 and 28 are entered frame-wise uninterruptedly.

[0016] The data read out to the X-bus 223 and Y-bus 224 are those stored at data memory addresses indicated to the input dual memories 211 and 212 by the address generators 221 and 222 that are controlled by the signals provided by the decoder 245 by decoding a 80-bit length horizon-type microcodes read out in accordance with the address of the command memory 243 indicated by the sequencer 242. The data placed on the X-bus 223 and Y-bus 224 are entered in parallel to the pipeline arithmetic unit 225, which implements a series of signal processing including coding and local decoding and outputs the result to the Z-bus 241. Among the process outputs placed on the Z-bus, the coded output is stored in the FIFO memory 232 and the local decoded output is stored in the FIFO memory 233 by way of the output bus 231.

[0017] The FIFO memories 232 and 233 are buffer memories of FIFO configuration. Feedback data consisting of the output data and local decoded data are read out of the output ports 29 and 30 at the read control timing for the assigned area produced from the video frame signal, and a piece of video frame local decoded data and coded output data in compliance with the scanning order are produced.

[0018] The data memory 226 which is controlled by the output of the address generator 227 is used by a work memory which is necessary for the process of the pipeline arithmetic unit 225 and a table which stores constants. The mode register 228 consists of a register file including registers for loading immediate values from the decoder 245.

[0019] This digital signal processing apparatus is principally based on the foregoing area division parallel processing, and is intended such that each signal processing module 20 deals with a divided frame area independently on a realtime basis. When the digital signal processing apparatus is intended for the achievement of a coder as shown in Fig. 5, only portions excluding the variable-length coder 256, video multiplexer 257, transmission buffer 258 and coding controller 270 can be realized. Namely, it is not suitable for a continuous process in one video frame, and is limited to the inter-frame coding loop process ranging from the input frame buffer 251 to the block identifier 253, coder 254, local decoder 260, coding frame memory 263, and to the motion compensator 265 useful for data completely divisible within a frame.

[0020] Since each signal processing module 20 implements the same process for each frame, the processing program stored in the instruction memory 243 can be a single program. When a frame is divided into M areas (M is an integer greater than or equal to 1), the number of process cycles Nc per pixel which can be dealt with on a realtime basis by one signal processing module 20 is given by the following calculation.

$$Nc = Mc \cdot Tf/Mp \cdot Np \qquad \text{(clocks/pixel)}$$

where Mc is the frequency of machine cycle (Hz), Tf is the frame period (sec), Mp is the number of horizontal pixels in the assigned area, and Np is the number of vertical pixels in the assigned area.

[0021] On this account, if a frame is divided into four areas, for example, each having the assignment of a signal processing module 20, the number of process cycles Nc is increased by four fold, and it becomes possible for the video signal processing, which is required to be very fast, to be dealt with on a realtime basis by an increased number of relatively slow signal processing modules 20.

[0022] The conventional digital signal processing apparatus arranged as described above have the following problems for processing video signals.

(a) For the achievement of very fast processing, a frame must be divided into numerous small areas, however, certain signal process algorithms do not allow independent processes for areas below a certain minimal division size. Therefore, realtime processing can not be achieved by increasing the parallelism.

(b) Because of a fixed distribution of load to signal processing modules, the process time must be set to meet the longest one when each signal processing module has a different process time. Therefore, the system has an unnecessarily increased parallelism relative to the processing capacity.

(c) Data input and data processing each take one frame time, and data input and output each need a 1-frame buffer memory, resulting in a longer time lag and an increased memory capacity. Therefore, the system involves a significant loop delay in feedback control and the like, and it is difficult to realize the coding controller 270 in Fig. 5 for example.

(d) Since the system is intended for a complete parallel processing, it cannot perform such a process as scanning the entirety of a same frame horizontally.

[0023] Fig. 6 is a block diagram of the conventional digital signal processing system disclosed in the proceeding (No. S10-1) of the 1986 annual convention of the communication department of The Institute of Electronics and Communication Engineers of Japan. In the figure, indicated by 31 is a dual-port internal data memory (will be termed 2P-RAM) capable of reading and writing two sets of data simultaneously, 32 is an address generator which calculates the address of read data or write data, 33 is a data bus used for the internal transfer of data related to computation, 34 and 35 are selectors which select data in the 2P-RAM 31, 36 is a register which holds computation data selected by the selector 34, 35 is a register which holds computation data selected by the selector 35, 38 is a multiplier, 39 is a register which holds the output of the multiplier 38, 40 is a selector which selects the output of the register 36 or accumulators (ACC0 - ACC3) 44, 41 is a selector which selects the output of the registers 39 or 37, 42 is an arithmetic/logic unit which performs computations for the outputs of the selectors 40 and 41, and 43 is a selector which selects the output of the arithmetic/logic unit 42 or data in an external data register 46. The accumulators 44 are used to hold the output of the arithmetic/logic unit 42 for cumulative computations. The external data register 46 is to hold data from an external data memory 47. Indicated by 45 is an external address register which holds address data provided by the address generator 32 and transfers it to the external data memory 47.

[0024] Next, the operation will be described. This signal processing system based on a digital signal processor performs command fetching and decoding for the preset microprogram, data reading, computation, and computation result writing, in a parallel pipeline processing mode. The following describes the operation of 3-input-1-output computation.

[0025] The arithmetic/logic unit, multiplier, address generator, data memories and selectors are controlled in the microcommand mode.

[0026] Arithmetic operations for two inputs, including addition, subtraction, maximum evaluation, minimum evaluation, etc. are expressed generically by $a \oplus b$, and a multiplication operation for two inputs is expressed generically by

a x b, where a and b are independent data.

[0027] The arithmetic operations and multiplication are combined to form 3-input-1-output operations, and they are defined by the following expressions.

$$Z_i^1 = (ai \oplus bi) \times ci \quad (1)$$

$$Z_i^2 = (ai \times bi) \oplus ci \quad (2)$$

where i = 1 to N, and ai, bi and ci are sets of independent data stored in the 2P-RAM 31.

[0028] Fig. 7 shows the sequence of process for implementing the 3-input operation of the form of expression (1) by the digital signal processing system, for example, shown in Fig. 6.

[0029] The data address generator 32 sets up the starting addresses for two data sets A and B, and selects the simple incremental mode. Then the two data sets A and B are loaded through the selectors 34 and 35 into the registers 36 and 37. The selectors 40 and 41 select the registers 36 and 37, respectively, so that the arithmetic/logic unit 42 implements the arithmetic operation ai⊕bi. The selector 43 selects the arithmetic/logic unit 42 to hold the operation result temporarily in one of accumulators (ACC0 - ACC3) 44, and the resultant data is sent over the data bus 33 and through the external register 46 and stored in the external memory 47, which addressing mode is the simple incremental mode because of it being linked to one of addresses for the 2P-RAM 31 in the address generator 32.

[0030] In the subsequent step ST3, the data address generator 32 sets up the starting addresses of the data set C and data set ai⊕bi, and ci data is read out of the 2P-RAM 31 to the register 36. The selector 35 selects the data bus to load the data of ai ⊕ bi in the external memory 47 into the register 37. In this case, in order to have a coincident timing of reading for the data set C and data set ai⊕bi, step ST4 needs to expend two cycles of useless command reading for the external memory in advance.

[0031] The two sets of data are rendered multiplication by the multiplier 38 in step ST5, and the result is stored in the register 39. In the next cycle, the resultant data is passed through the arithmetic/logic unit 42 and, after being held temporarily in one of the accumulators (ACC0 - ACC3) 44, transferred over the data bus 33 to the 2P-RAM 31.

[0032] These operations are carried out in parallel on the basis of the pipeline process, and the operations from the reading of 2P-RAM 31 until the storing of the process result in the external memory 47 for N pieces of data sets will take N + 3 machine cycles in the case of an arithmetic operation.

[0033] The steps of operations are listed in the following Table 1 and Table 2. Table 1 is for the operation of ai⊕bi and the transfer of the result to the external memory 47, and Table 2 is for the reading the resultant ai⊕bi from the external memory 47, the operation of (ai⊕bi)·ci , and the transfer of the result to the 2P-RAM. In both tables, symbol "x" represents an indefinite value. Storing in the external data register 46 completes in machine cycle N + 3 in both tables, and the external data register 46 is read uselessly in machine cycle 0 (two machine cycles) in Table 2.

Table 1

Machine cycle	Register 36	Register 37	Register 39	a c c x	External data register 46
1	a_1	b_1	x	x	x
2	a_2	b_2	$a_1 \times b_1$	$a_1 \oplus b_1$	x
3	a_3	b_3	$a_2 \times b_2$	$a_2 \oplus b_2$	$a_1 \oplus b_1$
4	a_4	b_4	$a_3 \times b_3$	$a_3 \oplus b_3$	$a_2 \oplus b_2$
⋮	⋮	⋮	⋮	⋮	⋮
N	a_N	b_N	$a_{N-1} \times b_{N-1}$	$a_{N-1} \oplus b_{N-1}$	$a_{N-1} \oplus b_{N-2}$
N + 1	x	x	$a_N \times b_N$	$a_N \oplus b_N$	$a_{N-1} \oplus b_{N-1}$
N + 2	x	x	x	x	$a_N \oplus b_N$
N + 3	x	x	x	x	x

EP 0 690 376 B1

Table 2

Machine cycle	Register 36	Register 37	Register 39	a c c x	External data register 46
0	x	x	x	x	$a_1 \oplus b_1$
1	$a_1 \oplus b_1$	c_1	x		$a_2 \oplus b_2$
2	$a_2 \oplus b_2$	c_2	$(a_1 \oplus b_1) \times c_1$		$a_3 \oplus b_3$
3	$a_3 \oplus b_3$	c_3	$(a_2 \oplus b_2) \times c_2$		$a_4 \oplus b_4$
⋮	⋮	⋮	⋮		⋮
N	$a_N \oplus b_N$	c_N	$(a_{N-1} \oplus b_{N-1}) \times c_{N-1}$		x
N + 1	x	x	$(a_N \oplus b_N) \times c_N$	x	x
N + 2	x	x	x		x
N + 3	x	x	x	x	x

[0034] Next, after two useless reading cycles of the external memory 47 for timing purposes, multiplication is carried out for N pieces of data sets and the results are stored in the 2P-RAM 31. These operations take N + 3 machine cycles,

which are added by two command cycles for address initialization, and a total of 2N + 10 cycles are expended. An operation of expression (2) also takes 2N + 10 cycles. Accordingly, it will be appreciated that if a 3-input-1-output operation is conducted for N pieces data sets using a processor with the ability of 2-input operation at most, it will take about 2N machine cycles (provided that N is sufficiently large).

[0035] The following describes the cumulative operation for the results of the foregoing 3-input-1-output computation.

$$S = \sum_{i=1}^{N} (ai \oplus bi) \times ci \qquad \ldots (3)$$

$$S = \sum_{i=1}^{N} (ai \times bi) \oplus ci \qquad \ldots (4)$$

In the case of expression (3), the multiplication result for ai⊕bi and ci (output of register 39) and the intermediate cumulative value are entered to the arithmetic/logic unit 42, and the result of summation is entered back to the same accumulator 44 through the selector 43. Thereby, the process takes 2N + 10 cycles unchanged.

[0036] In the case of expression (4), the data sets (ai x bi)⊕ci which have been stored temporarily in the 2P-RAM 31 are read out sequentially and summed by the arithmetic/logic unit 42, and therefore the process needs another N cycles, resulting in a total of 3N + 10 cycles.

[0037] The conventional digital signal processing system is formed as described above, and therefore for a 3-input-1-output operation of three independent data sets, it performs two times of 2-input-1-output operation. In addition, the process time is further extended for address control, memory transfer and other processes.

[0038] Fig. 8 is a diagram showing in brief the image coding transmitter which implements the conventional motion compensatory operation method disclosed in an article entitled "Dynamic Multistage Vector Quantization for Images", journal of The Institute of Electronics and Communication Engineers of Japan, Vol. J68-B, No. 1, pp. 68 - 76, Jan. 1985. In the figure, indicated by 1 is an input signal of image data formed of a plurality of consecutive frames on the time axis, 52 is a motion compensator which produces a prediction signal on the basis of the resemblance computation of correlation between the current frame represented by the input signal 1 and the previous frame represented by a previous frame signal 53 which is the previous reduced signal 1, 54 is motion vector information provided by the motion compensator 52 indicative of the position of a prediction signal block, 55 is a prediction signal produced by the motion compensator 52, 56 is a coder which codes the difference between the input signal 1 and prediction signal 56, 57 is a decoder which decodes the signal coded by the coder 56, and 58 is a frame memory which stores data reproduced through the summation of the signal from the decoder 57 and the signal from the motion compensator 52.

[0039] The performance of the foregoing arrangement will be described in connection with Fig. 9. The motion compensation process is to calculate for the input signal 1 the amount of distortion between a 11-by-12 block located in a specific position in the current frame shown in Fig. 9(A) and M pieces of blocks in the search range S in the previous frame shown in Fig. 9(B) to evaluate the position of the block y providing a minimal distortion relative to the position of the input block, i.e., motion vector V, and to recognize the signal of the minimal distortion block as a prediction signal.

[0040] The number of motion vectors V under search within the search range S in the given frame is assumed to be M (an integer greater than 1). The amount of distortion of the position of a specific motion vector V between the previous frame blocks and the current input block is calculated as a sum of absolute values of differences as follows.

$$di = \sum_{h=1}^{K} |yih - xh| \qquad \ldots (5)$$

where input vectors x = {x1, x2, ..., xk}, search object blocks yi = {yi1, yi2, ..., yik}, i = 1, 2, ..., M, and M and K are fixed values. The motion vector V is evaluated as follows.

$$V = Vi \{\min di \mid i = 1, 2, \ldots, M\} \qquad (6)$$

[0041] Fig. 10 shows the sequence of operations for detecting the motion vector V. Step STll calculates a distortion di at each of K pieces of sampling points on the basis of expression (5), and the next step ST12 compares the di with the minimal distortion D at position I, and, if di < D, the variables are replaced to be D=di and I = i. These operations are repeated for the number of search vectors, i.e., the operational process of expression (6), to determine the final minimal distortion D and its position I.

[0042] These operations must be completed within the period of each frame entered successively, and therefore a high-speed digital signal processor is required.

[0043] As an example, the digital signal processing system shown in Fig. 6 is used to carry out the motion compensation process. In this case, the multiplication-sum operation takes place K x M times for each input block, and the number of machine cycles is the total time expended by M times of processes including comparison and updating. Generally, the number of cycles for comparison and updating is small enough as compared with that of the multiplication-sum operation, and the volume of motion compensation operation for one block is virtually equal to K x M machine cycles.

[0044] However, since these operations are determined fror the time corresponding to the period of frames entered successively, parallel processing will be needed for the mass multiplication-sum operations to be performed in a short time, depending on the operation process cycle time of a particular digital signal processor.

[0045] The conventional motion compensation scheme is implemented as described above, and in order to ensure the operation time for an enormous volume of operations when carried out using a digital signal processor, the processor needs to have parallel processings, resulting in an increased complexity and scale of hardware structure.

[0046] The U.S. Patent 4,661,849 assigned to PicTel Corporation is concerned with an image sequence transmission method which contains estimating the motion displacement between successive image frames of a sequence. Each image is divided into nonoverlapping blocks, wherein each block including a plurality of picture elements. The method then includes successive selection of an initial value of a motion vector displacement for each successive block in accordance with a steepest descent estimation technique. Thereafter, successive values of the motion displacement vector for a block are determined according to the steepest descent approach and the iteration process is terminated when an iteration error measure, which is associated with the steepest descent method, is less than a predetermined threshold value. An adaptive approach is used wherein the step size parameter is incremented in accordance with a gradient error used in the steepest descent technique.

[0047] A scientific paper of Yasuo Ariki and Toshiyuki Sakai titled "Moving Picture Analysis Based on a Hierarchical Model Approach" which is published in "Systems-Computers-Controls", Vol. 12, No. 4, July 1981, Washington U.S., pages 37-45, is also concerned with an efficient processing method for a moving picture. According to this citation, for efficient selection of frames with significant changes, a small picture is divided into several blocks and correlations are calculated between corresponding blocks of different frames. If a block with correlation lower than the threshold is found, that frame is selected. This method uses a hierarchical structure model by selecting frames with smaller consecutive correlations. The hierarchical structure consists of feature points, boundary lines and regions, which shall make it possible to trace the feature point and boundary line based on prediction or the trace the whole object pattern.

SUMMARY OF THE INVENTION

[0048] The present invention is intended to overcome the foregoing prior art deficiencies.

[0049] The object of this invention is to provide a motion compensative operation method which, in constructing the motion compensator of an image coding system with a digital signal processing apparatus, requires less number of parallel processors, thereby enhancing the simplicity and compactness of the hardware structure.

[0050] According to the present invention the object is achieved by a motion compensative operation method wherein a current input frame of digital image data consisting of a olurality of frames entered successively is divided into a plurality of blocks, and, in detecting a block which provides a minimal distortion resulting from computation of resemblance between patterns based on the cumulation of differential absolute values or differential square values of pixels in block between a block of the current input frame and blocks of M in number of the previous frame, a maximum of K pixels are used for the pattern resemblance computation, intermediate checks are conducted n times during the resemblance computation at time points when the number of reference pixels is smaller than K, a block is determined to be outside the range of comparison for finding a minimal distortion block if a cumulative value at each time point is greater than a threshold value preset for each intermediate check time point, and a block having a minimal distortion is detected by computation from among blocks whose cumulative values are below the threshold values at all time points of intermediate checks, wherein M is a positive integer, K is an integer greater than or equal to one and smaller than or equal to the number of a total number of pixels in one block and n is an integer greater than or equal to one.

BRIEF DESCRIPTION OF THE DRAWINGS

[0051]

Fig. 1 is a block diagram showing the multiprocessor system of a conventional digital signal processing apparatus;
Fig. 2 is a diagram explaining the assigned areas of the processors shown in Fig. 1;
Fig. 3 is a block diagram showing the arrangement of other conventional digital signal processing apparatus;
Fig. 4 is a block diagram showing in detail the arrangement of the signal processing module shown in Fig. 3;
Fig. 5 is a block diagram showing the algorithm of the high-efficiency coder for a moving image;
Fig. 6 is a block diagram showing the arrangement of a third conventional digital signal processing apparatus;
Fig. 7 is a flowchart showing the process of 3-input arithmetic operation using the digital signal processing apparatus shown in Fig. 6;
Fig. 8 is a block diagram showing in brief the arrangement of the image coding transmitter which carries out the conventional motion compensative operation method;
Fig. 9 is a diagram used to explain the conventional motion compensative operation method;
Fig. 10 is a flowchart showing the operational process for detecting a motion vector in the conventional motion compensative operation method;
Fig. 11 is a block diagram showing the digital signal processing apparatus based on the first embodiment of this invention;
Fig. 12 is a diagram explaining the area assignment for the processors shown in Fig. 11;
Fig. 13 is a block diagram showing the arrangement of the digital signal processing apparatus formed by connecting in cascades a plurality of digital signal processors (DSP blocks) shown in Fig. 11;
Fig. 14 is a diagram showing the concept of process of each DSP block shown in Fig. 13;
Fig. 15 is a block diagram showing the digital signal processing apparatus based on the second embodiment of this invention;
Fig. 16 is a block diagram showing the internal arrangement of the signal processor shown in Fig. 15;
Fig. 17 is a diagram explaining the concept of control operation of the digital signal processing apparatus shown in Fig. 15;
Fig. 18 is a diagram explaining the relation between parameter data and processing block data in the digital signal processing apparatus shown in Fig. 15;
Fig. 19 is a diagram showing the correspondence between data blocks and a frame;
Fig. 20 is a block diagram of the arrangement in which a plurality of digital signal processors are included in the digital signal processing apparatus shown in Fig. 15;
Fig. 21 is a block diagram showing the digital signal processing apparatus based on another embodiment of this invention;
Fig. 22 is a flowchart showing the operational process of the digital signal processing apparatus shown in Fig. 21;
Fig. 23 is a flowchart showing an embodiment of the inventive motion compensative operation method using a digital signal processing apparatus;
Fig. 24 is a diagram used to explain the method of intermediate check for the computation of distortion in the inventive motion compensative operation method; and
Fig. 25 is a diagram showing the arrangement of pixel samples at sampling points in a block according to the intermediate check method for the distortion computation.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0052] Specific embodiments of the present invention will now be described with reference to the drawings.

[0053] Fig. 11 shows, as an embodiment of this invention, an example of the image coder of the digital signal processing apparatus. In the figure, input data 1 is entered to a first through third input memories 6. A task controller 7 estimates the number of valid pixels on the basis of the contents of the input memory 6, determines the distribution of coding process among a first, second and third DSPs 2, and issues control signals as address control signals 8 to the DSPs 2. Upon receiving the address control signals 8, the first, second, and third DSPs 2 issue addresses 9 to respective first, second and third input memories 6 to fetch data 10 assigned for processing, and implement the coding processes based on the preset program. Upon completion of processes, the first, second and third DSPs 2 store processed data in an output memory 11, which, after reading the whole data of the DSP block, sends the processed data to the next DSP block.

[0054] In this case, each DSP 2 is controlled by the task controller 7 so that all DSP 2 have even numbers of valid pixels assigned, and therefore the image coding process time is controlled so that the difference of process times among the DSPs 2 is minimal. Namely, in case of coding an image with numbers of valid pixels as shown in Fig. 12

(b), an area A having a relatively small number of valid pixels is enlarged to A', an area C having a relatively large number of valid pixels is also enlarged to C', and an area B having a larger number of valid pixels is reduced to B', as shown in Fig. 12(a), by the task controller 7. The task controller 7 issues the address control signals 8 corresponding to the assignment distribution to the first, second and third DSPs 2.

[0055] For example, in response to the issuance of the address control signal 8 for coding the image data of area A to the first DSP 2, it produces the address 9 for the area A' in the first input memory 6 to fetch data and implements the image coding process by following the prescribed program. Similarly, the second and third DSPs 2 are directed to carry out the image coding processes for the areas B' and C', respectively. Consequently, the first, second and third DSPs 2 have their numbers of valid pixels EA', EB' and EC' for coding virtually made even, i.e., the same quantity of image data to be processed, as shown in Fig. 12(b). As a result, the maximum volume of process M' dealt with by the inventive apparatus becomes sufficiently less than that M of the conventional apparatus, and the process time required for each DSP block is reduced.

[0056] Fig. 13 shows the inter-frame coder constructed by a serial connection of DSP blocks in three stages. Each DSP block performs the process shown in Fig. 14. The first DSP block 12 enters upon the input data 1 and, after producing a differential signal, implements the valid/invalid judgment, evaluates the distribution of the numbers of valid pixels in the image data, and sends the information to the task controller 7. Based on the information, the task controller 7 issues address control signals 8 for dictating such address adjustment that the DSPs in the second DSP block 13 have even assignments of data. Each DSP in the second DSP block 13 implements the process by adjusting the read address as described above. The third DSP block 14 is designed to operate identically.

[0057] Although in the foregoing embodiment the DSP process assignment areas are controlled on the basis of the valid pixel distribution among areas in image data, the present invention is not confined to this scheme, but feedback DSP assignment control based on the general quantity distribution of transmitted information is also possible, for example.

[0058] A second embodiment of this invention will be explained with reference to the drawings. Fig. 15 shows an example of the configuration of a digital signal processing apparatus, the second embodiment of this invention. In the figure, 301 is a data flow control section (D F C) working as a control means; 302 are control parameter data output from the data flow control section 301; 303 is a common memory (C M) which stores feedback data, a large capacity data and table, etc.; 304 is a task table (T B) which stores a processing status of each signal processor element (P E) 318; 305 is a common bus (C-BUS) which has the function as a status communicating means consisting of at least a bus connected to the common memory 303, the task table 304 and each signal processor element 318; 306 is a video frame synchronizing signal (F p) which discriminates the starting point of a video frame to be supplied to the data flow control section 301 in the case of inputting video signals etc.; 307 are feedback data (F b) which inform the data flow control section 301 of the occupying status, data quantity of a sending buffer etc. and finishing of one frame data processing etc. output from an output control section 308 described later; 308 is an output control section (O C) provided with a buffer memory for outputting data at a certain constant speed in restructuring processed blocks output from a plurality of signal processor elements (P E) 318 for example in the scanning order in a video frame; 309 is an input terminal of analog signals; 310 is an A/D converter; 311 are digitized input data; 312 is a parameter memory (P M) consisting of dual port memories; 313 is an input frame buffer consisting of dual port memories for functioning as a block formation means by memorizing input data 311 temporarily; 314 is a bus connecting the parameter memory 312 to the signal processor elements 318; 315 is a bus connecting the input frame buffer 313 to the signal processor elements 318 in order to supply data in a block unit; 316 is a common bus input/output port connected to the common bus 305; 317 is an interruption control port for sending/receiving timing control signals from the data flow control section 301; 318 are individual signal processor elements (P E) and these signal processor elements are provided with software which functions as a starting means, and these signal processor elements are mutually connected with buses 314 and 315, and said last stage signal processor element 318 and the output control section 308 are also connected with buses 314 and 315; 319 is an output terminal through which data are output at a certain constant speed and timing from the output control section 308; 320 is a multiprocessor module comprising the parameter memory 312, the input frame buffer 313 and a plurality of signal processors 318 connected in series through the buses 314 and 315, for example.

[0059] The data flow control section 301 has a judgment means which scans the task table 304 at a certain constant cycle and judges the processing conditions of individual signal processor elements 318. The data flow control section 301 also has a control means which based on the result of the judgment means decides if each signal processing module can process the next signal process block and when the processing is found to be possible it makes process start by sending out an interruption signal to the interruption control port 317 and when the processing is found to be impossible it instructs the transfer of the signal process block to another signal processing module which can process the block. When a parallel processing of a constant cycle, in which the task table 304 is scanned, is to be done the scanning period shall be the number of parallelness times of the input cycle of the signal process block, and when a series processing is to be done the scanning period shall be 1/n of the input cycle; thus by the synchronization with

the input data frame (for example a video frame) the matching with the real time can be maintained.

[0060] Fig. 16 shows an example of the internal elements constitution of the signal processor elements 318 as shown in Fig. 15. In the figure, 330 is a terminal to which the common bus input/output port 316 is to be connected; 331 is a terminal to which the interruption control port 317 is to be connected; 332 is a terminal to which the buses 314 and 315 are to be connected; 333 is similarly a terminal which connects the buses 314 and 315 between the adjacent signal processors; 334 is an external bus control section (BUS-CONT) with the function as a competitive control means to control the make/break of the common bus 305 through the bus 316; 335 is a bus for loading a writable command memory (W C S) 336, which memorizes a signal processing program, from the external bus control section 334 at an initial time; 337 is a BUSREQ which requires the connection of the common bus 305 to the external bus control section 334; 338 is a BUSACK which denotes the permission for the BUSREQ 337; 339 are command codes which are successively read out from the writable command memory 336 according to the signal processing program; 340 is a digital signal processor (D S P) which executes data processing; 341 is an INTACK which informs an interruption control section (INTER-CONT) 345 of the reception of an interruption from the digital signal processor 340; 342 is, on the contrary to it, an INTREQ which informs the digital signal processor 340 of the requirement of an interruption; 343 is a bus to connect an internal bus 344 to the common bus 305 through the external bus control section 334, and the internal bus 344 is directly connected to the digital signal processor 340; 345 is an interruption control section (INTR-CONT) which processes an interruption signal from the data flow control section 301; 346 is a bus which writes the parameter of a processed data block on a dual port memory 349 through the internal bus 344; 347 is similarly a bus which writes processed block on the dual port memory 349; 348 is a bus which connects a work memory in the dual port memory 349 and the internal bus 344; 349 is a dual port memory provided with a parameter memory, data memory and work memory which outputs data to the adjacent signal processor element 318 through the terminal 333 and buses 314 and 315.

[0061] Fig. 17 explains the internal control operation of the digital signal processing apparatus shown in Fig. 15, and the same parts as those shown in Fig. 15 are given the same symbols; the explanation of them is therefore omitted.

[0062] In the figure, 351 is a block which shows analytical operation of a parameter inside the signal processor element 318; 352, 353, 354 are blocks which show the operation of individual signal processing subroutines A, B and C according to the parameter of each of them; 355 is a block which shows the contents of a parameter memorized in the dual port memory 349; 356 is a block which shows the contents D of processed block data memorized in the dual port memory 349.

[0063] Fig. 18 explains an example of the relation between the parameter data and process block data until a data block is successively given a series of function processes and an output result is obtained through series and parallel processes of block units executed in the digital signal processing apparatus shown in Fig. 15. In the figure, 360 is a block address (B A D) showing the position of an input block in a frame; 361 is a processing number (PN) showing the kind of a process to be given to said block; 362 is a flag (PFLG) which discriminates the result of the process; 363 is a data block in which for example eight subblocks are combined to form a block.

[0064] Fig. 19 shows an example of correspondence between the data block 363 shown in Fig. 18 and one video frame when a picture coding process is performed in this system. In the figure, 365 is one video frame; 366 is a data block when a picture is divided into 16 lines x 16 pixels; 367 is a subblock which is obtained when the block is further divided into 8 blocks of 4 lines x 4 pixels.

[0065] An explanation of the operation based on Fig. 15 is given in the following. Input data 311 digitized by an A/D converter 310 are memorized in an input frame buffer 313 being scanned in a raster form in synchronization with a video frame synchronizing signal 6, for example. Input data 311 memorized in the input frame buffer 313 are added to initial parameter data 302 by the data flow control section 301 by blocks and the parameter data 302 are memorized in the parameter memory 312. These parameter memory 312 and input frame buffer 313 consist of dual port memories and writing/reading is simultaneously possible between two independent ports.

[0066] Data blocks are read from the input frame buffer 313, and the parameter is read in a data block unit from the parameter memory 312. Data blocks and parameter are sent through the buses 314 and 315 to the signal processor 318 element where they are given the first process of a series of functional processes in a block unit. Next, the results and the rewritten parameters are written in the dual port memory 349 in the signal processor element 318. It is the basic function of a processor module 320 to execute processes successively between the adjacent signal processor elements 318 and to execute a pipeline processing for each block unit.

[0067] When a processing is executed for each block unit, if a feedback data such as coded previous frame data are to be referred to, feedback data are input to the common memory 303 connected to the common bus 305 and memorized. The process of a new video frame is performed by such processing that the other signal processor 318 than the one which data have written through common bus 305 refers the common memory 303. If the writing of the feedback data of the previous frame is not completed in the proper position in the common memory 303, the execution time of the process shall be specified.

[0068] When the processing of a unit (block processing) is finished, each signal processor element 318 memorizes

the status showing the completion of the present processing in the task table 304, and wait the next processing. The data flow control section 301 scans the task table 304 and when the processing of the former stage signal processor element 318 is completed, it sends out an interruption signal to said signal processor element 318 and start the next processing. By repeating the operation, the execution of the operation control of each signal processor element 318 is performed.

[0069] To conduct parallel processing in a block unit for each processor module 320, the data processing condition in the input frame buffer 313 of each processor module 320 is detected with the status information of the initial stage signal processor 318 and individual block data are distributed by proper load distribution and input to each multi-processor module 320.

[0070] These results are shown by the control parameter data of the initial stage and the signal processor element 318 discriminates the processing for the block by deciphering the above results and executes a proper processing. Among these processings there are for example functional processors such as a block identifier 253, a coder 254, a local decoder 260, an inter-frame subtracter 252, a motion compensator 265, an inter-frame adder 261, a variable length coder 256, and besides them a processing which performs only load distribution such as a processing of transferring block data is included.

[0071] In the data flow control section 301, it is possible to make an arbitrary signal processor 318 undertake an arbitrary processing by controlling the first stage parameter; owing to such performance as mentioned above the load can be so distributed to signal processor elements 318 as to make them work efficiently as much as possible.

[0072] The output control section 308 reconstitutes processed blocks which are output at random times into for example a scanning order of an input video frame and produces a resultant output for an output terminal 319 and also produces feedback data 307 to inform the data flow control section 301 of these data.

[0073] The output control section 308 takes charge for example of a video multiplex section 257 and a transmitting buffer 258 shown in Fig. 5, and it outputs a feedback signal 269 from the transmitting buffer 258 to a coding control section 270 which takes charge of the data flow control section shown in Fig. 15.

[0074] The data flow control section 301 takes charge of the functions of above-mentioned load distribution and the coding control section 270 as shown in Fig. 5, and finds the block identification control signal 273 and coding control signal 274 and multiplex them in the control parameter data for the execution of the whole characteristic control. Refer to Fig. 16; the processing of a single signal processor element 318 is started by the interruption from the data flow control section 301, and the contents of the parameter memory 312 is input to it through an internal bus 344. On the basis of the discrimination result of the contents, the processing of one unit of block data is performed by a digital signal processor 340.

[0075] The result and rewritten parameters are written in a dual port memory 349, and the status is set in the task table 304 through an external bus control section 334; thus the preparation for the next process is ready. An interruption control section 345 interfaces the interruption from the data flow control section 301 with the digital signal processor 340. The parameter and the data written in the dual port memory 349 are read by an adjacent signal processor element 318 which is connected to a terminal 333, and the next stage process is given.

[0076] Fig. 17 shows the flow of these processes performed by the data flow control section 301, and it shows the relation between the control of writing/referring of feedback data to the common memory 303 and the control of status writing in the task table 304 by the data flow control section 301 through the common bus 305, and the start processing control in the signal processor element 318 by a parameter analyzer 351.

[0077] Fig. 18 shows the rewriting of the contents of control parameter data 302, which are added corresponding to an input block data 363, and the flow of these processes. A block address which shows for example the position in a frame or time sequential order of a block, and a flag 362 which is referred to on the kind of the next process and the contents of the next process are contained in the control parameter data 302. The block address 360 is used for the discrimination of a special process in a certain case for example with an end point in a picture or for the restructure of data in the output control section 308 when a process is finished. The flag 362 shows for example the results etc. of coding control information 271, a block identification control signal 273, coding control signal 274, and a block identifier 253 as shown in Fig. 5. Input block data 363 are set to have the minimum size handled in a unit processing. The motion compensator 265 shown in Fig. 5 has a block of 16 x 16 size and after the block identifier 253 blocks of 4 x 4 sizes are handled. In such a case as mentioned above where a block size differs for each unit processing, block sizes are arranged to have matching between a maximum block size and a subblock size contained in it. In this case, eight pieces of 4 x 4 blocks are combined to constitute a 16 x 16 block. When coding of a picture is performed, this block corresponds to a small picture element made by dividing an ordinary one frame into small square picture elements.

[0078] Fig. 19 shows an example where one video frame 365 is divided into a block 366 and subblocks 367.

[0079] In the above embodiment, a signal processor element 318, which has a single digital signal processor 340, is shown but when a higher speed processing is preferable a hierarchical structure combined with a plural number of digital signal processors can be used. The constitution of the signal processor element 318 in the case of the hierarchical structure is shown in Fig. 20. In this case, as the load for the data flow control section 301 increases a local data flow

control section 370, a local common memory 371, and a local task table 372 are provided inside the signal processor 318 in order to locally execute the optimum load distribution inside the signal processor. The data flow of the digital signal processor 340 which is connected to a local common bus 373 is the same as that shown in Fig. 15 except that the operation is executed inside the signal processor 318.

[0080] In the above embodiment, a series/parallel structure is adopted but in some case a complete parallel or complete series structure is effective according to the purpose of a signal processing and a real time processing could be possible.

[0081] The other embodiment of this invention is explained with reference to Fig. 21. In Fig. 21, 420, 421 and 422 are address generators for readout data; 423 is an address generator for writing data; 424, 425 and 426 are data memories, and address data generated by the address generator 423 are input to these memories; 427, 428 and 429 are data buses which transfer readout data from the data memories 424, 425 and 426; 430, 431 and 432 are registers for holding data transferred from data buses 427, 428 and 429; 433 is a register to hold the output of the register 432; 434 is a selector to select the output of the register 430 or that of the register 433; 435 is a selector to select the output of the register 431 or that of the register 441; the selector 434 and the selector 435 constitute a first selector group; 436 is a selector to select the output of the register 430 or the output of a register 439; 437 is a selector to select the output of the register 431 or the output of the register 433; the selector 436 and the selector 437 constitute a second selector group; 438 is an operator which operates by inputting the output of the selectors 434 and 435; 440 is a multiplier which performs multiplication by inputting the output of selectors 436 and 437; the register 439 is the one to hold the output of the operator 438; a register 441 is the one to hold the output of the multiplier 440; 442 is a selector which selects the input from the register 439 or the input from the register 441 and outputs it; 443 is an adder which adds the output of the output selector 442 and the output of an accumulator 444 and outputs to said accumulator 444; 445 is a data bus to transfer output data of the accumulator 444 and the output selector 442; 446 is an interface circuit which performs outputting/inputting of data to/from external circuits; 451 - 453, 461 - 463, 471 - 473 denote signal lines which output the output of data memories 424, 425 and 426 to data buses 427, 428 and 429.

[0082] The following are the explanation of operation. In Fig. 21, assume that data series with N elements, $\underline{A}$ = (aili = 1 to N), $\underline{B}$ = (bili = 1 to N), $\underline{C}$ = (cili = 1 to N) are previously stored respectively in the data memory 424, data memory 425, and data memory 426.

[0083] Under the conditions above, the operation when the operation of three inputs and one output is performed is shown below. The operation processing flow is shown in Fig. 22.

[0084] To begin with, at a step ST31, top addresses of three series of input data and of an output result storing memory are initially set by address generators 420, 421 and 422. After that the address generators are assumed to take simple increment actions.

[0085] The data memory 424 corresponds to the address generator 420; the data memory 425 corresponds to the address generator 421; the data memory 426 corresponds to the address generator 422. Individual data memories 424, 425 and 426 readout data based on the addresses of address generators 420, 421 and 422.

[0086] Data are input to three data buses 427, 428 and 429 (X-BUS, Y-BUS, Z-BUS) respectively from data memories 424, 425 and 426, so that for the outputting of each of these data memories 424, 425 and 426 to a specified data bus, only one bus out of three is controlled to be effective, and the other two are controlled to be in the state of a high impedance. In this case, the output of data buses is limited to that of the one which is made to be effective. For example, when $\underline{A}$ data series is to be input to the register 430, the $\underline{A}$ series data are output to the signal line 451, and the signal lines 461 and 471, which output data from other data memories 425 and 426 to the data bus 427, are in the state of a high impedance. The same thing goes for other data buses.

[0087] Each of these data series are set respectively in the registers 430, 431 and 432. Three data buses 427, 428 and 429 can select data from three data memories 424, 425 and 426, so that 3^3 kinds of data set combinations can be supplied to the registers 430, 431 and 432.

[0088] Two expressions as shown below are defined in the way of three input operation and then the processing method is shown in the following:

$$(ai \oplus bi) \times ci \tag{7}$$

$$(ai \times bi) \oplus ci \tag{8}$$

where $(x \oplus y)$ expresses an arithmetic and logic operation for finding results or values of addition, subtraction, maximum values or minimum values for two input data x, y, and $(x \times y)$ expresses multiplication. The explanation of operation processing flow of the expression (7) is given in the Table 3. The mark of "X" in the table represents an unknown.

Table 3

Machine cycle	Register 430	Register 431	Register 432	Register 433	Register 439	Register 441	Output of output selector 442
1	a_1	b_1	c_1	x	x	x	x
2	a_2	b_2	c_2	c_1	$a_1 \oplus b_1$	x	x
3	a_3	b_3	c_3	c_2	$a_2 \oplus b_2$	$(a_1 \oplus b_1) \times c_1$	x
4	a_4	b_4	c_4	c_3	$a_3 \oplus b_3$	$(a_1 \oplus b_2) \times c_2$	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
N	a_N	b_N	c_N	c_{N-1}	$a_{N-1} \oplus b_{N-1}$	⋮	⋮
N + 1	x	x	x	c_N	$a_N \oplus b_N$	⋮	
N + 2	x	x	x	x	x	$(a_N \oplus b_N) \times c_N$	
N + 3	x	x	x	x	x	x	$(a_N \oplus b_N) \times c_N$

EP 0 680 376 B1

[0089] At a step ST32 a selector 434 selects the side of a register 430 and a selector 431 selects the side of a register 435. By the use of these two selected data (ai and bi) the operation (ai ⊕ bi) is performed with an operator 438, and the result is stored in a register 439. This value is output from the register 439 in the next step.

[0090] The data ci in the register 432 are delayed by the register 433 by one step. In the next step a selector 436 selects the side of the register 439 and a selector 437 selects the side of a register 433. By the use of these two data, (ai ⊕ bi) is multiplied by ci with the multiplier 440 and the result (ai ⊕ bi) x ci is stored in a register 441. This value is output from the register 441 in the next step. By an output selector 442's selecting the register 441, the data (ai ⊕ bi) x ci are sent to one of the data memories 424, 425 and 426 through a data bus 445 based on the address shown by the address generator 423.

[0091] In this invention, readout of data, execution of operation and writing of data are continuously executed by a pipeline processing, so that the control of each section can be operated in parallel. Therefore if the three input one output operation is executed for a data series with N elements, from the time when the first datum is readout until the time when the processing result of the last datum is written into a memory, the period of (N + 3) cycles are required.

[0092] The explanation of operation processing flow of expression (8) is given in Table 4. The mark "x" in Table 4 represents an unknown.

Table 4

Machine cycle	Register 430	Register 431	Register 432	Register 433	Register 439	Register 441	Output of output selector 442
1	a_1	b_1	c_1	x	x	x	x
2	a_2	b_2	c_2	c_1	x	$a_1 \times b_1$	x
3	a_3	b_3	c_3	c_2	$(a_1 \times b_1) \oplus c_1$	$a_2 \times b_2$	x
4	a_4	b_4	c_4	c_3	$(a_2 \times b_2) \oplus c_2$	$a_3 \times b_3$	$(a_1 \times b_1) \oplus c_1$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
N	a_N	b_N	c_N	c_{N-1}	⋮	⋮	⋮
N + 1	x	x	x	c_N	⋮	$a_N \times b_N$	⋮
N + 2	x	x	x	x	$(a_N \times b_N) \oplus c_N$	x	⋮
N + 3	x	x	x	x	x	x	$(a_N \times b_N) \oplus c_N$

[0093] The operation in which three input data are readout to registers 430, 431 and 432 is the same as that in the case of expression (7). When the operation of expression (8) is executed, the selector 436 selects the side of the register 430 and the selector 437 selects the side of the register 431, and the operation (ai x bi) is performed by the multiplier 440 and the result is set in the register 441.

[0094] In the next step, the selector 434 selects the side of the register 433 and the selector 435 selects the side of the register 441, and the operation (ai x bi) ⊖ ci is executed by the operator 438 and the result is set in the register 439. In the next step, by the selector 442's selecting the side of the register 439 the selection result is written into one of the data memories 424 to 426.

[0095] Thus the case of the operation of expression (8) is the same as the case of expression (7), thereby the total processing time requires (N + 3) cycles.

[0096] In the case of the operation of two input one output, the value of (ai⊖bi) can be obtained through the procedure as shown in the following: the selector 434 selects the side of the register 430 and the selector 435 selects the side of the register 431 and after the operation is executed by the operator 438 the side of the register 439 is selected by the selector 442 in the next step. The value of (ai x bi) can be obtained through the procedure as shown in the following: the selector 436 selects the side of the register 430 and the selector 437 selects the side of the register 431, and after the execution of the operation with the multiplier 440 the selector 442 selects the side of the register 441 in the next step.

[0097] The processing speed in the case of three input one output is (2N + 10/N + 7) times of that of prior art, that is almost half times if N is a large number.

[0098] When a cumulative value is to be found in the three input one output operation, a cumulative value till a point on the way or an initial value is stored in the accumulator 444 and each one of the successive operation results is added to the cumulative value in the accumulator 444 with the adder 443 and the added result is stored in the accumulator 444 again. These processes are performed repeatedly. Processing cycles therefore are not increased due to cumulative operation.

[0099] Fig. 23 shows a flow chart to realize a method for motion compensative operation which refers to an embodiment of this invention. Fig. 24 is a drawing for the explanation of an intermediate check method in the distortion quantity operation in this invention. Fig. 25 is a disposition drawing of a pixel sample at a sample point in a block in the intermediate check method for distortion operation in this invention.

[0100] Before the operation process, on the first block among M pieces of candidate blocks for search in the previous frame data, distortion quantity of all the pixels in the block shall be measured; the distortion quantity in this case is defined to be the minimum distortion. As for the distortion quantity, differential absolute value sum is adopted. In the distortion quantity operation about on and after the second block the calculation of differential absolute values of all pixels is not needed, but at an intermediate check point if the intermediate distortion quantity exceeds a certain value, it is judged that the ultimate distortion quantity of the block cannot be smaller than the minimum distortion D and the distortion quantity operation for the residual part is stopped.

[0101] A block which gives the minimum distortion is detected by the calculation of the degree of approximation between the patterns by using the difference and accumulation of pixels in the respective M blocks which are selected out of the present input frame and the previous input frame (M is a positive integer). The number of pixels used for the calculation of the degree of approximation is K at the maximum (K is an integer greater than or equal to one and smaller than or equal to the number of a total number of pixels in one block). During the calculation of the degree of approximation at the time when the number of pixels in reference is less than K intermediate checks are performed four times, and an intermediate check point shall be provided in each 1/4 sample point. Fig. 25 shows examples of sample points used for distortion quantity operation. The mark O expresses a first time sample point for distortion quantity operation; the mark x expresses a second time sample point for distortion quantity operation; the mark Δ expresses a third time sample point for distortion quantity operation; the mark ⓞ expresses a fourth time sample point for, distortion quantity operation.

[0102] In Fig. 24 when the total number of sample points is assumed to be K, express threshold levels at a first, second and third intermediate check points as d1', d2' and d3'; then put

$$d1' = D/4 + th1 \tag{9-1}$$

$$d2' = D/2 + th2 \tag{9-2}$$

$$d3' = 3D/4 + th3 \tag{9-3}$$

where th1, th2, th3 can be set independently. Express the distortion quantity at the first, second and third intermediate

points as di1, di2 and di3.

[0103] In this case, di1 expresses the value of the first time distortion quantity in Fig. 25; di2 expresses a cumulative value, di1 plus the second time distortion quantity; di3 expresses a cumulative value, di2 plus the third time distortion quantity. Therefore, the cumulative value in which the fourth time distortion quantity is accumulated becomes the distortion quantity in which all sample points are included.

[0104] On the basis of a distortion quantity judgment at an intermediate check point if a block is estimated to have a large distortion quantity, the checking of the block is canceled before the block reaches the last check point to save useless operation processes. In other words if a distortion quantity di1 which is obtained by a distortion calculation at 1/4 K sample point in a step ST41 is found to be di1 > d1' by the judgment in the next step ST42, the block is canceled, if not, the operation is continued to the next step ST43 and the operation of distortion quantity di2 is performed with the distortion calculation at 1/2 K sample point. If it is found that di2 > d2' in the judgment in the next step ST44 the block is canceled and if not, the operation is continued into the step ST45. The operation of distortion quantity di3 is performed with the calculation at 3/4 K sample point, and if it is found that di3 > d3' by the judgment in the next step ST46 this block is canceled, if not, the operation is continued into the step ST47 and the distortion quantity di at K sample point is calculated for performing comparison and renewal.

[0105] As shown in the above if the processing is performed till the last step the same result is obtained as that obtained with the conventional method in which the whole pixels are used for a distortion operation. If the distortion quantity di, in this case, is smaller than the minimum distortion D, the value of the minimum distortion D is renewed for di and the motion vector index is renewed for the index i. The final minimum value of distortion D and the vector index I which shows the movement to give D can be obtained by repeating such operating processes as mentioned above by the number of times corresponding to the number of searching vectors till the process proceeds up to the Mth block.

[0106] In the above embodiment, the example where differential absolute value sum is used for distortion quantity operation is shown, but differential square value sum can also be used.

[0107] In the above embodiment, explanation is made about the case of motion compensative operation, but the execution of inner product vector quantization operation is also possible and the same effect can be obtained. When an operation result is compared with a threshold value at an intermediate check point, the relation in magnitude is opposite to what is mentioned in the above embodiment.

Claims

1. A motion compensative operation method wherein a current input frame of digital image data consisting of a plurality of frames entered successively is divided into a plurality of blocks, and, in detecting a block which provides a minimal distortion resulting from computation of resemblance between patterns based on the cumulation of differential absolute values or differential square values of pixels in block between a block of the current input frame and blocks of M in number of the previous frame, a maximum of K pixels are used for the pattern resemblance computation, intermediate checks are conducted n times during the resemblance computation at time points when the number of reference pixels is smaller than K, a block is determined to be outside the range of comparison for finding a minimal distortion block if a cumulative value at each time point is greater than a threshold value preset for each intermediate check time point, and a block having a minimal distortion is detected by computation from among blocks whose cumulative values are below the threshold values at all time points of intermediate checks, wherein M is a positive integer, K is an integer greater than or equal to one and smaller than or equal to the number of a total number of pixels in one block and n is an integer greater than or equal to one.

Patentansprüche

1. Bewegungs-kompensierendes Betriebsverfahren, bei welchem ein gegenwärtig eingegebenes Vollbild aus Digital-Bilddaten aus mehreren, aufeinanderfolgend eingegebenen Vollbildern in mehrere Blöcke aufgeteilt wird und beim Erfassen eines Blockes, welcher eine minimale Verzerrung bereitstellt, die aus einer Berechnung der Ähnlichkeit zwischen Mustern, basierend auf der Kumulierung differentieller Absolutwerte oder differentieller Quadratwerte von Bildpunkten im Block zwischen einem Block des gegenwärtig eingegebenen Vollbildes und einer Anzahl von M Blöcken des vorherigen Vollbildes resultiert, wird ein Maximum von K Bildpunkten zur Muster-Ähnlichkeits-Berechnung verwendet, Zwischen-Prüfungen werden während der Ähnlichkeits-Berechnung zu Zeitpunkten, wenn die Anzahl von Bezugs-Bildpunkten kleiner als K ist, n-mal ausgeführt, ein Block wird als außerhalb des Vergleichs-Bereiches zum Finden eines Minimal-Verzerrungs-Blockes bestimmt, wenn ein kumulativer Wert zu jedem Zeitpunkt größer als ein für jeden Zwischen-Prüf-Zeitpunkt voreingestellter Schwellwert ist, und ein Block mit einer minimalen Verzerrung wird durch Berechnung aus Blöcken erfaßt, deren kumulative Werte zu allen Zeitpunkten

der Zwischen-Prüfungen unterhalb des Schwellwertes sind, wobei M eine positive ganze Zahl ist, K eine ganze Zahl größer oder gleich Eins und kleiner oder gleich der Anzahl der Gesamtanzahl der Bildpunkte in einem Block ist und n ist eine ganze Zahl größer oder gleich Eins.

Revendications

1. Procédé de compensation de mouvement dans lequel une trame d'entrée en courant de données d'image numériques constituée d'une pluralité de trames entrées les unes à la suite des autres est divisée en une pluralité de blocs et, lors de la détection d'un bloc qui confère une distorsion minimale résultant d'un calcul de ressemblance entre des motifs basé sur la somme de valeurs absolues différentielles ou de valeurs carrées différentielles de pixels dans un bloc entre un bloc de la trame d'entrée en courant et un nombre M de blocs de la trame précédente, un maximum de K pixels est utilisé pour le calcul de ressemblance des motifs, des vérifications intermédiaires sont effectuées n fois au cours du calcul de ressemblance à des moments ponctuels lorsque le nombre de pixels de référence est inférieur à K, un bloc est déterminé comme étant hors de la plage de comparaison pour trouver un bloc de distorsion minimal si une valeur cumulée à chaque moment ponctuel est supérieure à une valeur de seuil prédéfinie pour chaque moment ponctuel intermédiaire de vérification et un bloc ayant une distorsion minimale est détecté par un calcul entre des blocs dont les valeurs cumulées sont inférieures aux valeurs de seuil à tous les moments ponctuels des vérifications intermédiaires, dans lequel M est un entier positif, K est un entier supérieur ou égal à un et inférieur ou égal au nombre total de pixels dans un bloc et n est un entier supérieur ou égal à un.

FIG.1

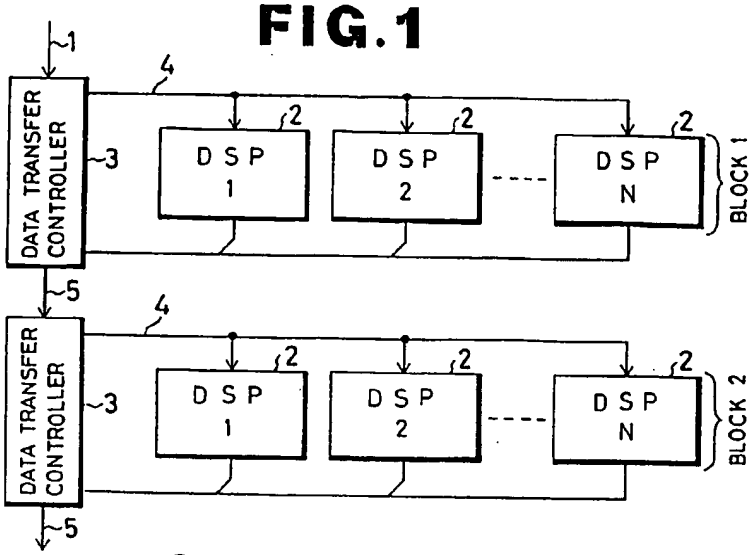
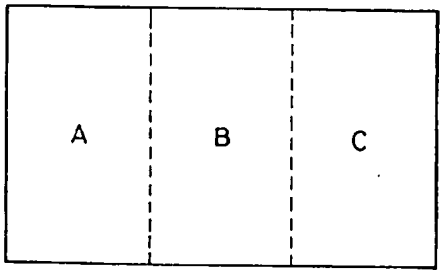


FIG.2

(a) DIVISION OF MEMORY AREA



(b) DISTRIBUTION OF NUMBERS OF VALID PIXELS IN EACH AREA

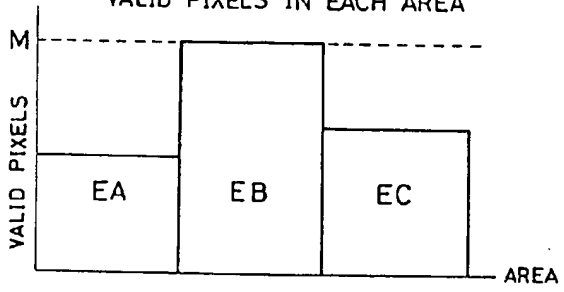


FIG. 3

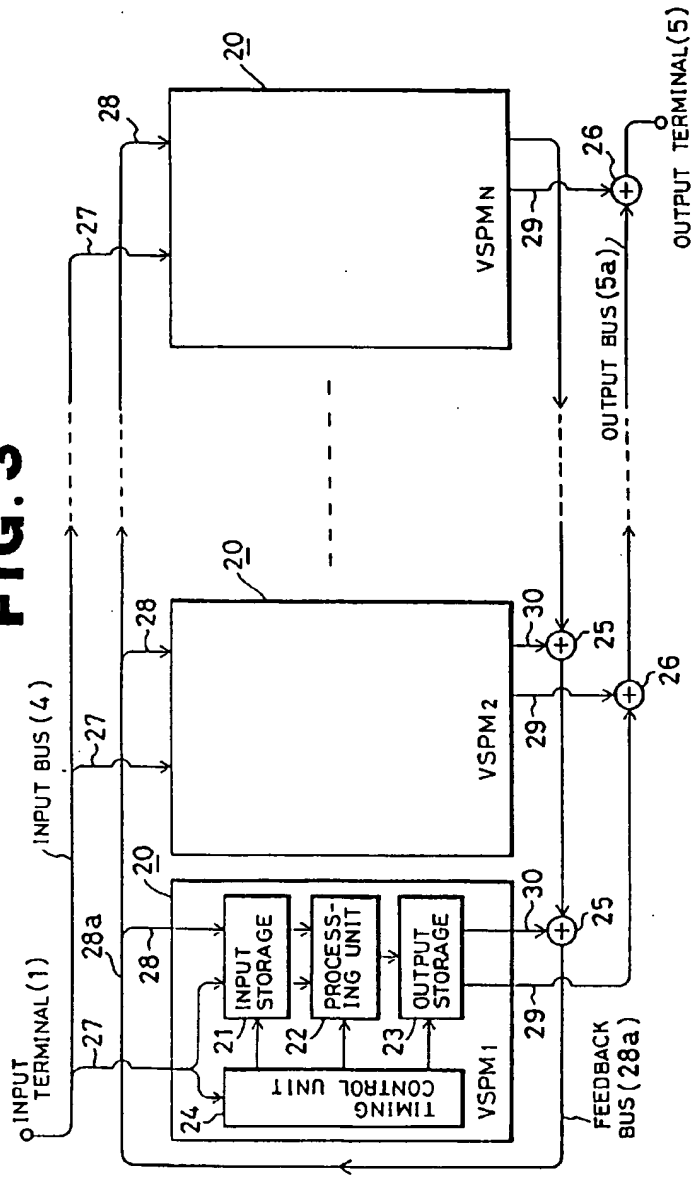

INPUT TERMINAL (1)
INPUT BUS (4)
VSPM 1
VSPM 2
VSPM N
INPUT STORAGE
PROCESS-ING UNIT
OUTPUT STORAGE
TIMING CONTROL UNIT
FEEDBACK BUS (28a)
OUTPUT BUS (5a)
OUTPUT TERMINAL (5)
20
21
22
23
24
25
26
27
28
28a
29
30

FIG.4

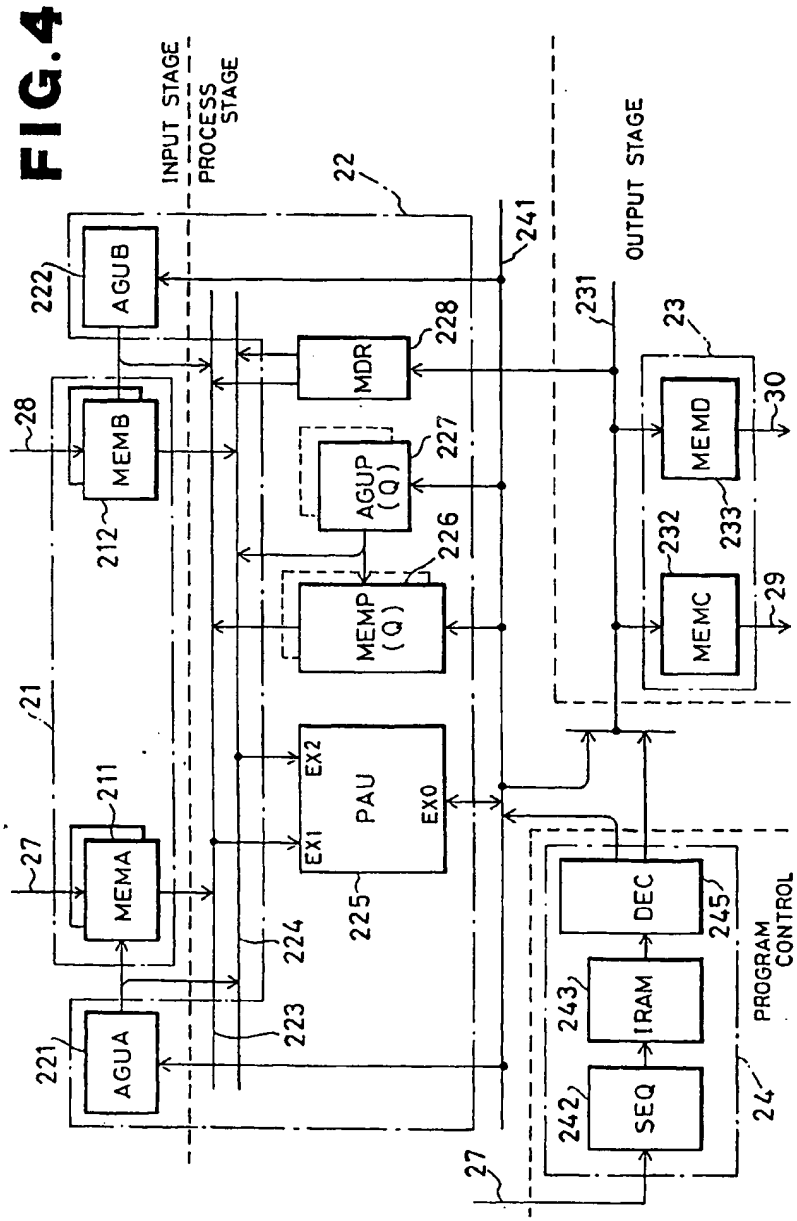

INPUT STAGE
PROCESS STAGE
OUTPUT STAGE
PROGRAM CONTROL
AGUA
AGUB
MEMA
MEMB
PAU
EX0
EX1
EX2
MEMP (Q)
AGUP (Q)
MDR
MEMC
MEMD
DEC
IRAM
SEQ
21
211
212
221
222
223
224
225
226
227
228
22
23
231
232
233
24
241
242
243
245
27
28
29
30

FIG. 5

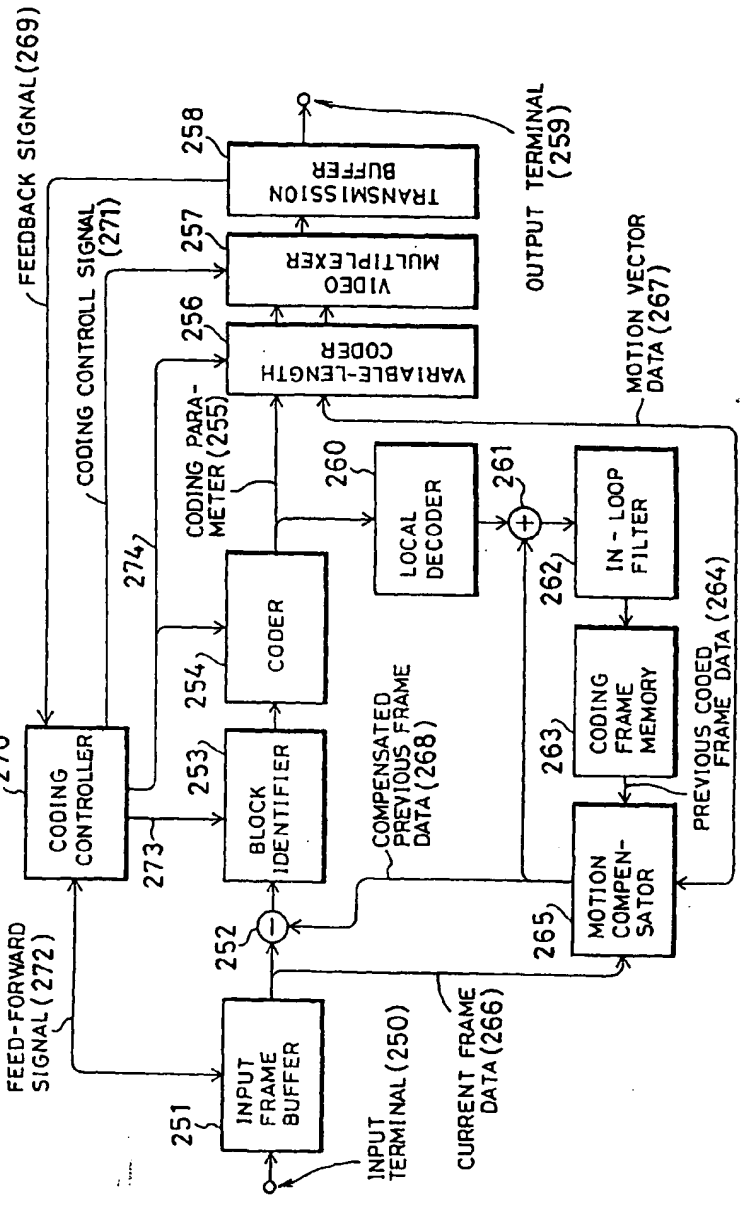

INPUT TERMINAL (250)
INPUT FRAME BUFFER
251
FEED-FORWARD SIGNAL (272)
CODING CONTROLLER
270
FEEDBACK SIGNAL (269)
CODING CONTROLL SIGNAL (271)
252
BLOCK IDENTIFIER
253
273
CODER
254
274
CODING PARA-METER (255)
VARIABLE-LENGTH CODER
256
VIDEO MULTIPLEXER
257
TRANSMISSION BUFFER
258
OUTPUT TERMINAL (259)
LOCAL DECODER
260
261
IN - LOOP FILTER
262
CODING FRAME MEMORY
263
MOTION COMPEN-SATOR
265
CURRENT FRAME DATA (266)
COMPENSATED PREVIOUS FRAME DATA (268)
MOTION VECTOR DATA (267)
PREVIOUS CODED FRAME DATA (264)

FIG. 6

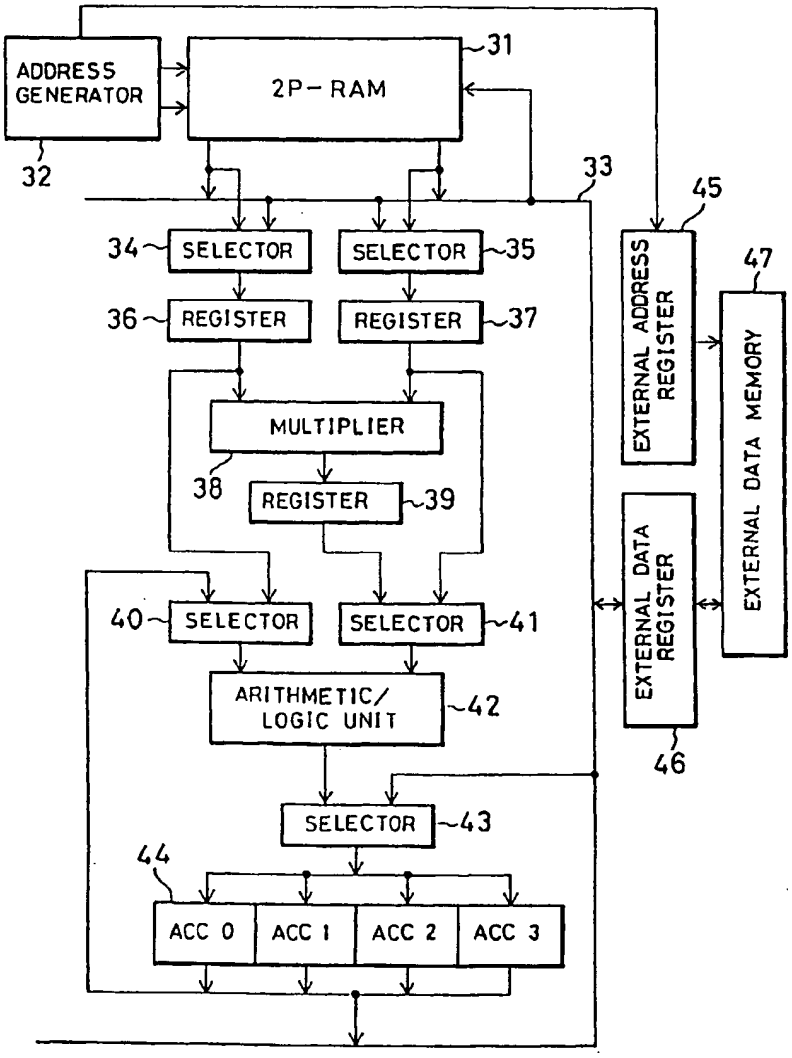

ADDRESS GENERATOR
32
2P-RAM
31
33
45
47
34
SELECTOR
SELECTOR
35
36
REGISTER
REGISTER
37
MULTIPLIER
38
REGISTER
39
EXTERNAL ADDRESS REGISTER
EXTERNAL DATA MEMORY
EXTERNAL DATA REGISTER
40
SELECTOR
SELECTOR
41
ARITHMETIC/ LOGIC UNIT
42
46
SELECTOR
43
44
ACC 0
ACC 1
ACC 2
ACC 3

FIG.7

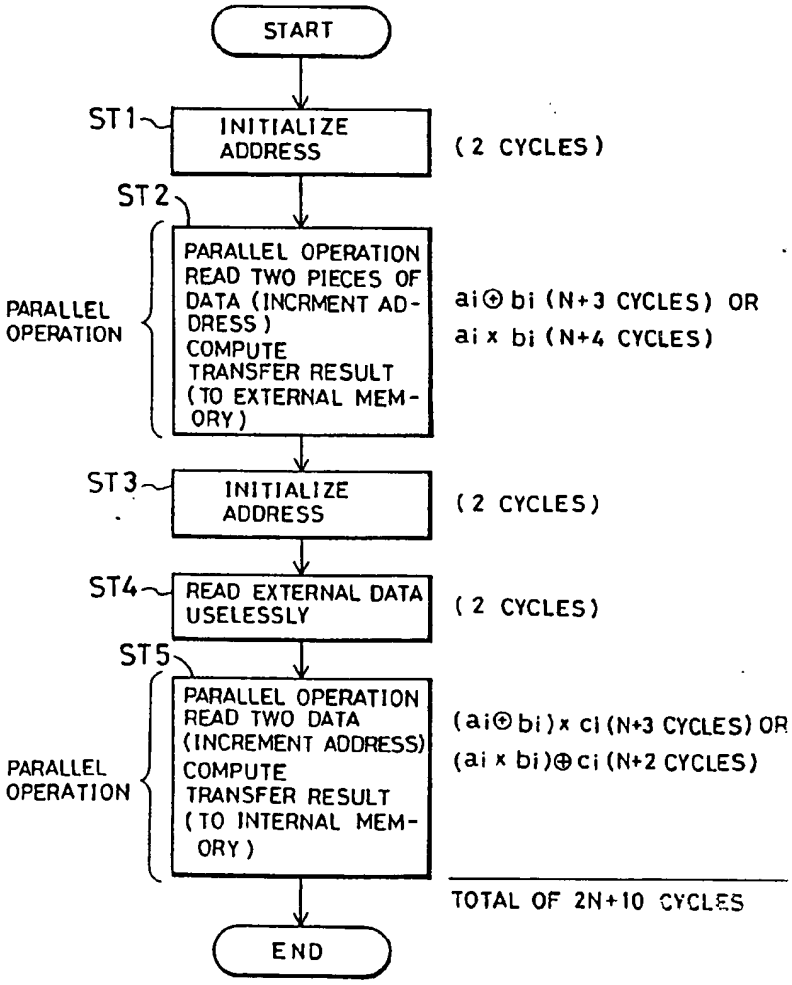

START
ST1
INITIALIZE ADDRESS
(2 CYCLES)
ST2
PARALLEL OPERATION
PARALLEL OPERATION READ TWO PIECES OF DATA (INCRMENT ADDRESS) COMPUTE TRANSFER RESULT (TO EXTERNAL MEMORY)
ai⊕bi (N+3 CYCLES) OR
ai x bi (N+4 CYCLES)
ST3
INITIALIZE ADDRESS
(2 CYCLES)
ST4
READ EXTERNAL DATA USELESSLY
(2 CYCLES)
ST5
PARALLEL OPERATION
PARALLEL OPERATION READ TWO DATA (INCREMENT ADDRESS) COMPUTE TRANSFER RESULT (TO INTERNAL MEMORY)
(ai⊕bi)x ci (N+3 CYCLES) OR
(ai x bi)⊕ci (N+2 CYCLES)
TOTAL OF 2N+10 CYCLES
END

FIG. 8

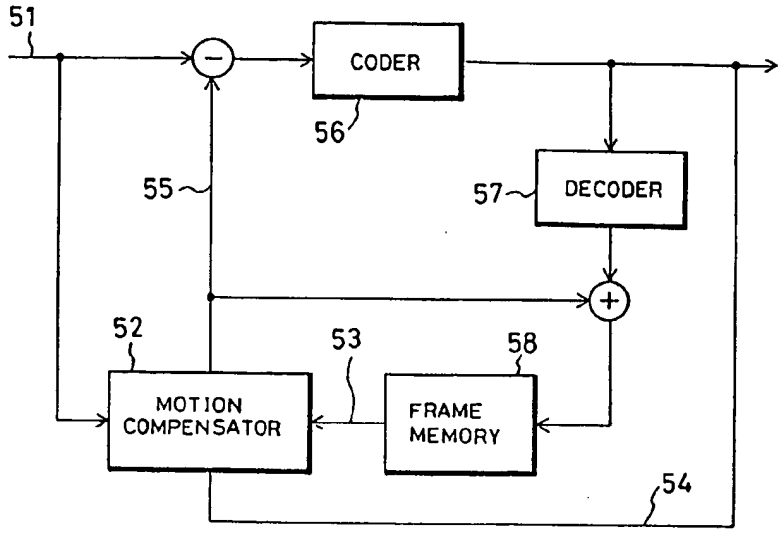

51
CODER
56
55
57
DECODER
52
53
58
MOTION COMPENSATOR
FRAME MEMORY
54

FIG. 14

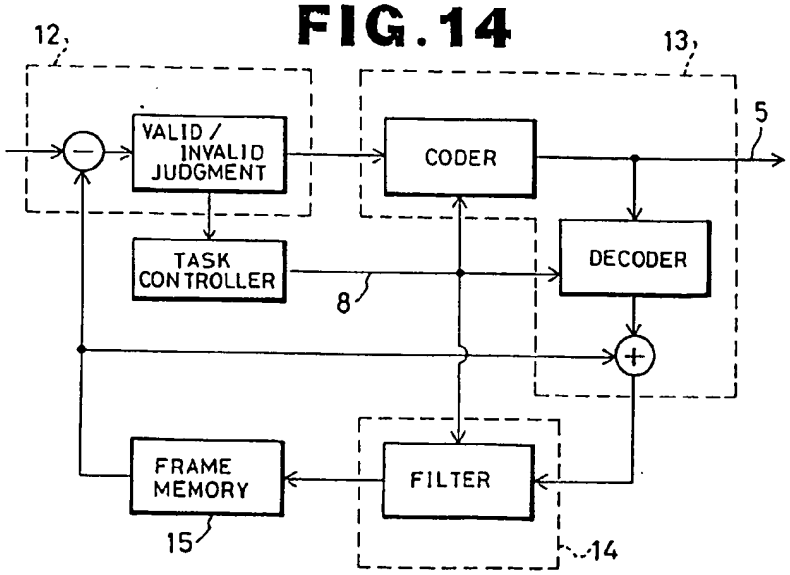

12
13
5
VALID / INVALID JUDGMENT
CODER
TASK CONTROLLER
8
DECODER
FRAME MEMORY
15
FILTER
14

FIG. 9

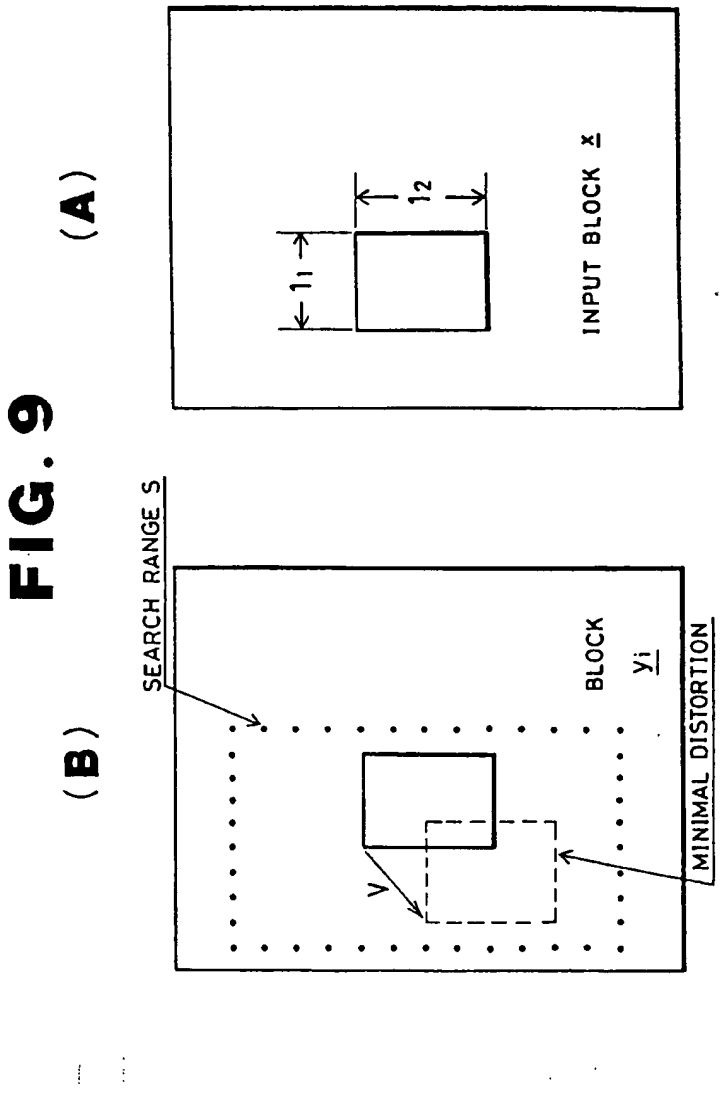

(A)
INPUT BLOCK x
l1
l2
(B)
SEARCH RANGE S
BLOCK
yi
v
MINIMAL DISTORTION

FIG.10

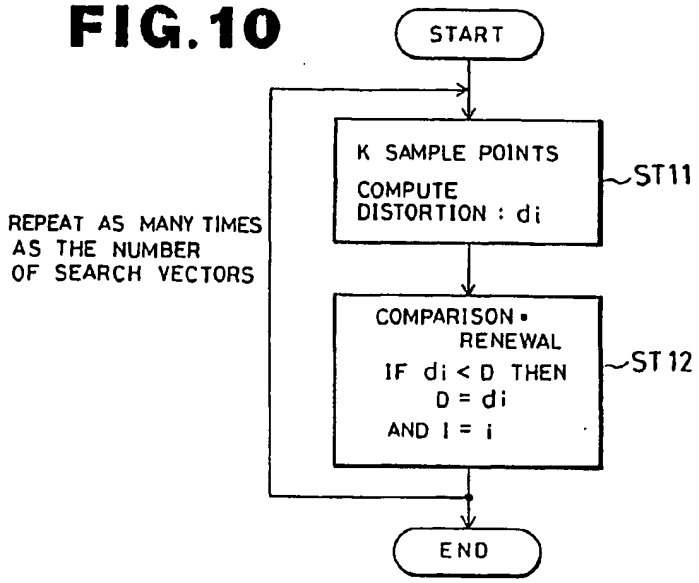

START
K SAMPLE POINTS
COMPUTE
DISTORTION : di
ST11
REPEAT AS MANY TIMES
AS THE NUMBER
OF SEARCH VECTORS
COMPARISON •
RENEWAL
IF di < D THEN
D = di
AND I = i
ST12
END

FIG.11

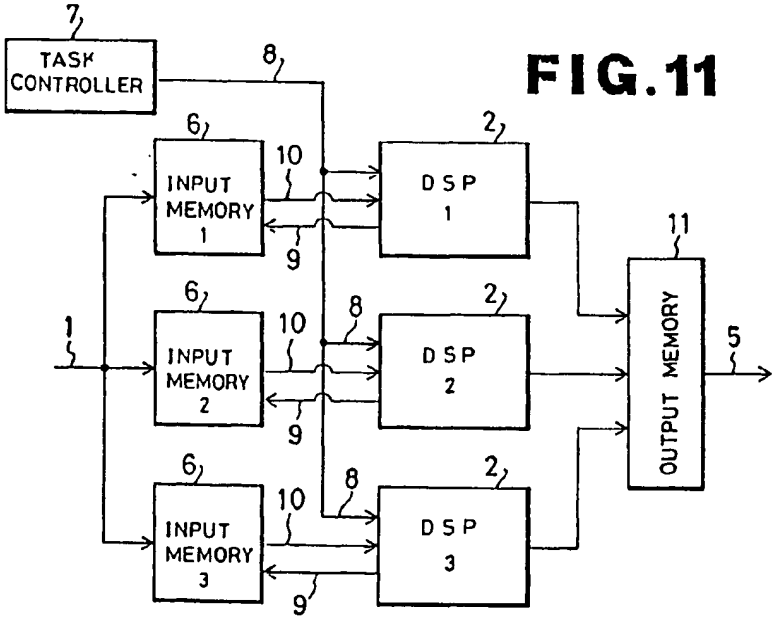

7
TASK
CONTROLLER
8
6
INPUT
MEMORY
1
10
9
2
DSP
1
11
OUTPUT MEMORY
5
1
INPUT
MEMORY
2
DSP
2
INPUT
MEMORY
3
DSP
3

FIG. 12

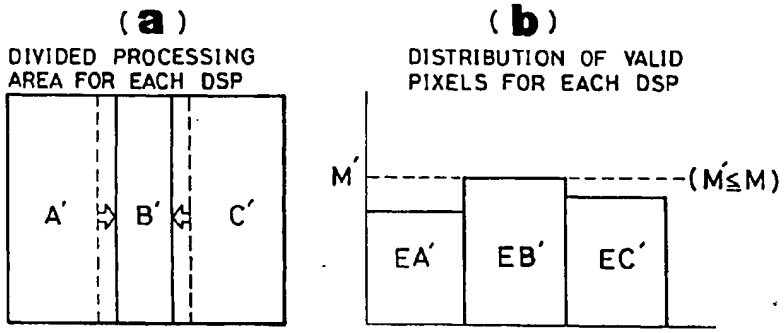

(a)
DIVIDED PROCESSING AREA FOR EACH DSP
A'
B'
C'
(b)
DISTRIBUTION OF VALID PIXELS FOR EACH DSP
M'
(M'≤M)
EA'
EB'
EC'

FIG. 13

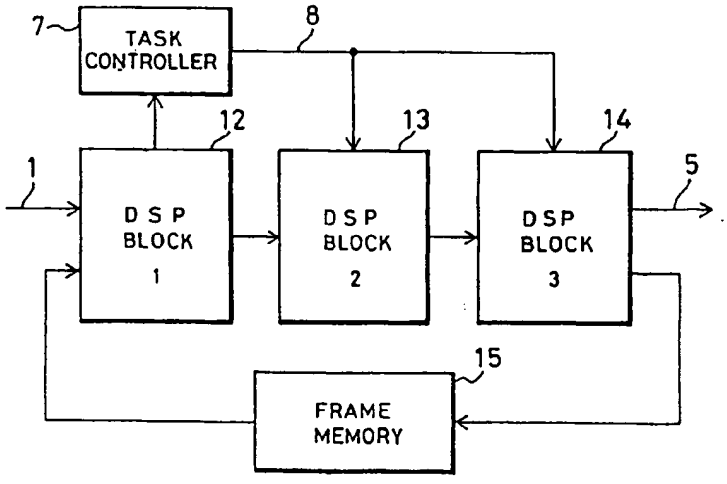

7
TASK CONTROLLER
8
12
13
14
1
DSP BLOCK 1
DSP BLOCK 2
DSP BLOCK 3
5
15
FRAME MEMORY

FIG. 15

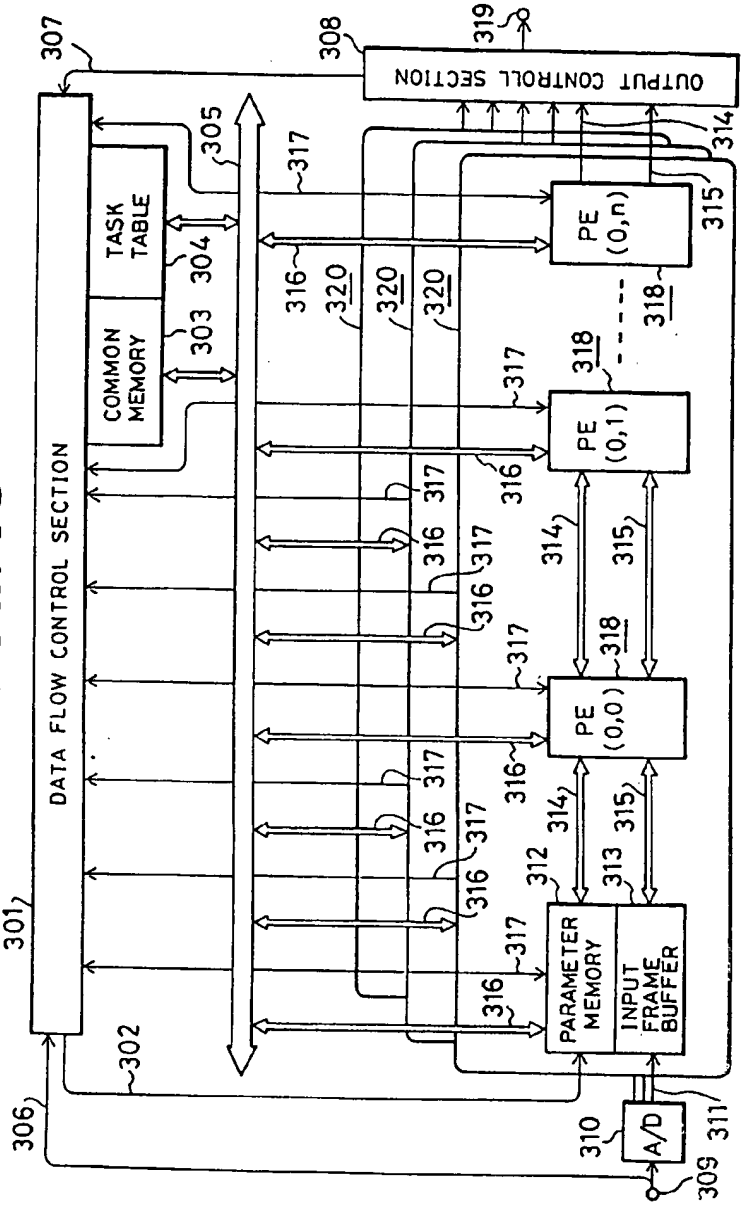

DATA FLOW CONTROL SECTION
COMMON MEMORY
TASK TABLE
OUTPUT CONTROL SECTION
PE (0,n)
PE (0,1)
PE (0,0)
PARAMETER MEMORY
INPUT FRAME BUFFER
A/D
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320

FIG.16

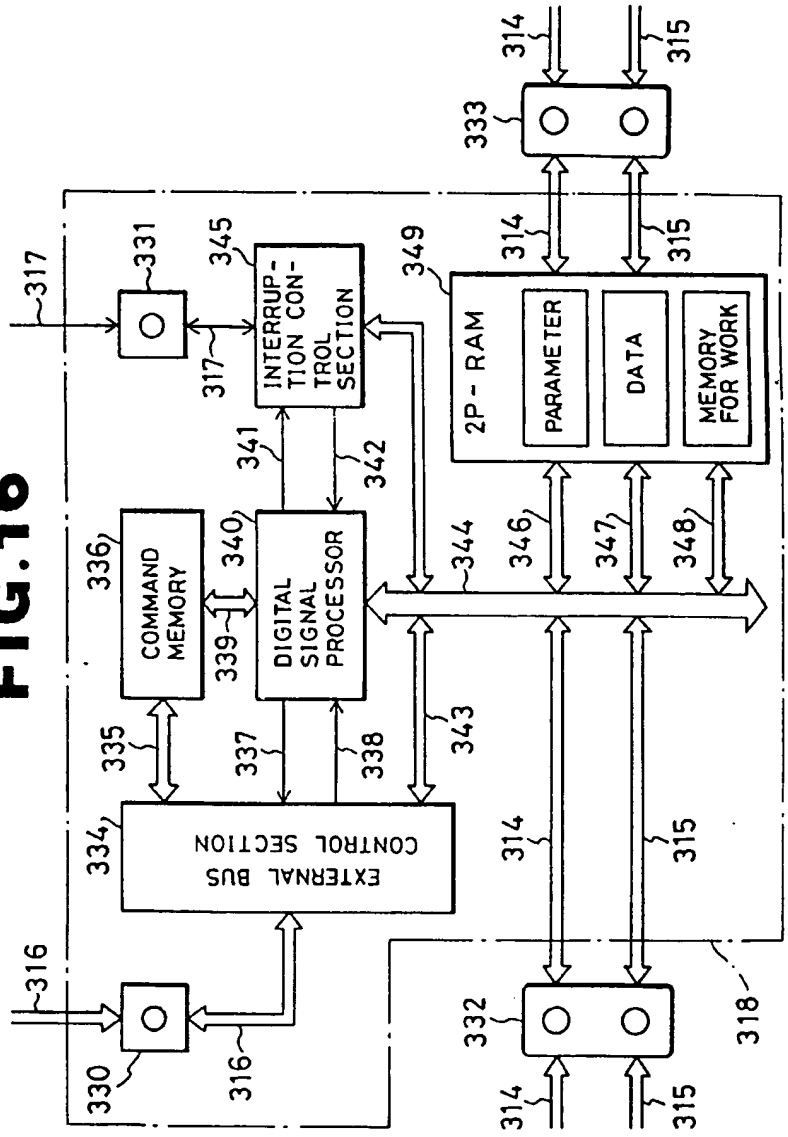

314
315
333
317
331
345
INTERRUP-
TION CON-
TROL
SECTION
349
2P-RAM
PARAMETER
DATA
MEMORY
FOR WORK
341
342
336
COMMAND
MEMORY
340
339
DIGITAL
SIGNAL
PROCESSOR
344
346
347
348
335
337
338
343
334
EXTERNAL BUS
CONTROL SECTION
316
330
332
318

FIG. 17

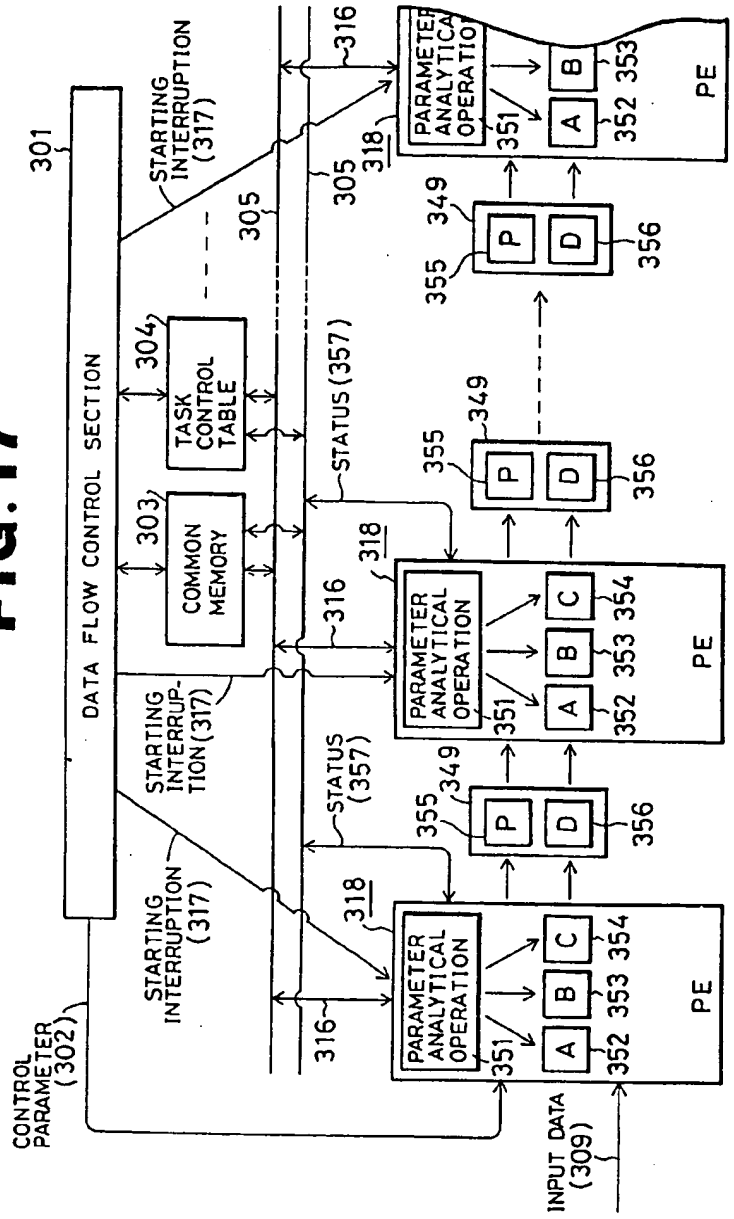

301
DATA FLOW CONTROL SECTION
STARTING INTERRUPTION (317)
304
TASK CONTROL TABLE
303
COMMON MEMORY
305
316
318
STATUS (357)
PARAMETER ANALYTICAL OPERATION
PE
A
B
C
351
352
353
354
349
P
D
355
356
CONTROL PARAMETER (302)
INPUT DATA (309)

FIG.18

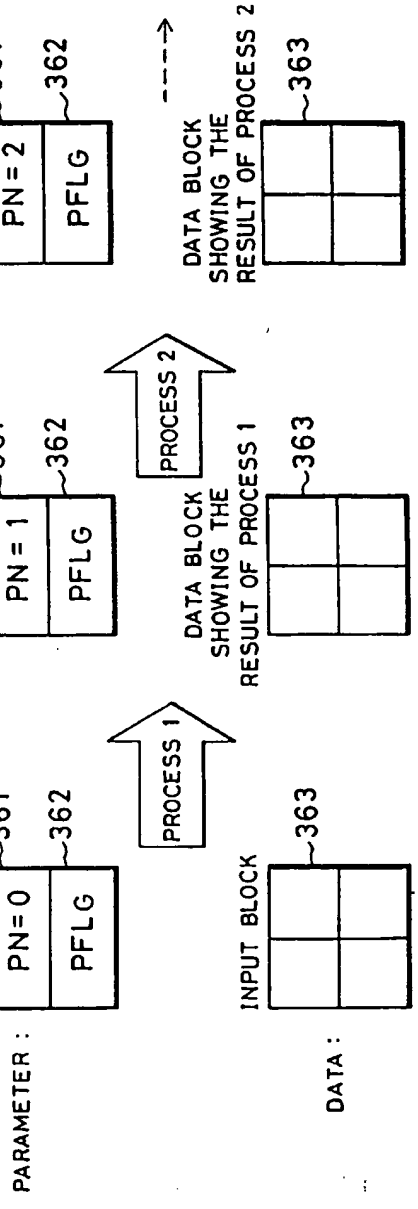

360
361
362
BAD
PN=0
PFLG
PARAMETER :
PROCESS 1
INPUT BLOCK
363
DATA :
360
361
362
BAD
PN=1
PFLG
DATA BLOCK SHOWING THE RESULT OF PROCESS 1
363
PROCESS 2
360
361
362
BAD
PN=2
PFLG
DATA BLOCK SHOWING THE RESULT OF PROCESS 2
363

FIG.19

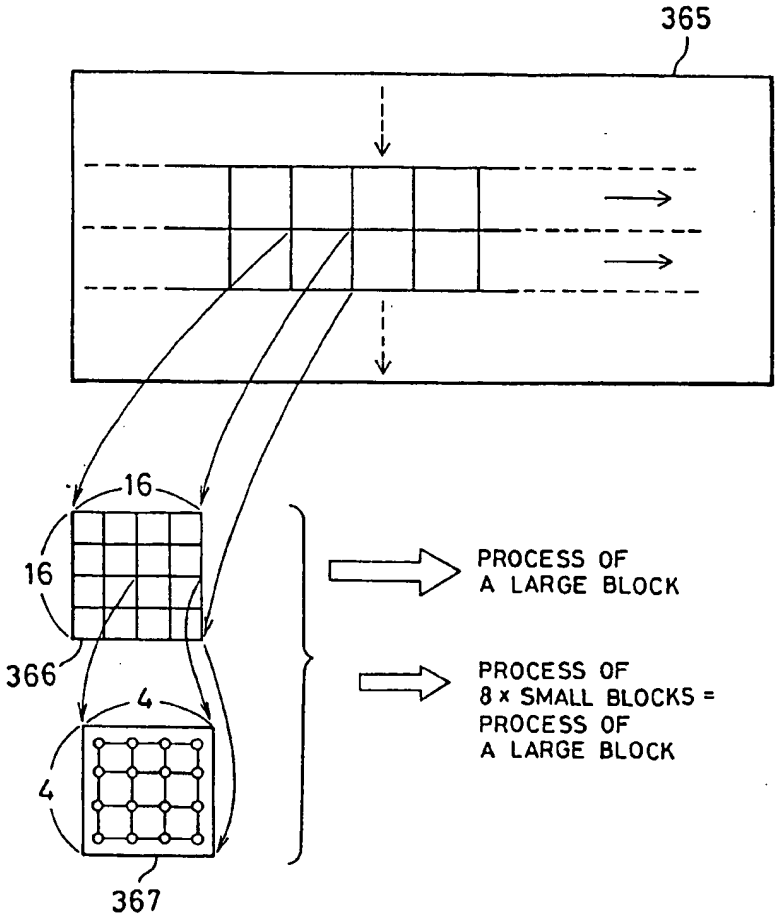

365
16
16
366
4
4
367
PROCESS OF
A LARGE BLOCK
PROCESS OF
8 x SMALL BLOCKS =
PROCESS OF
A LARGE BLOCK

FIG. 20 (I)

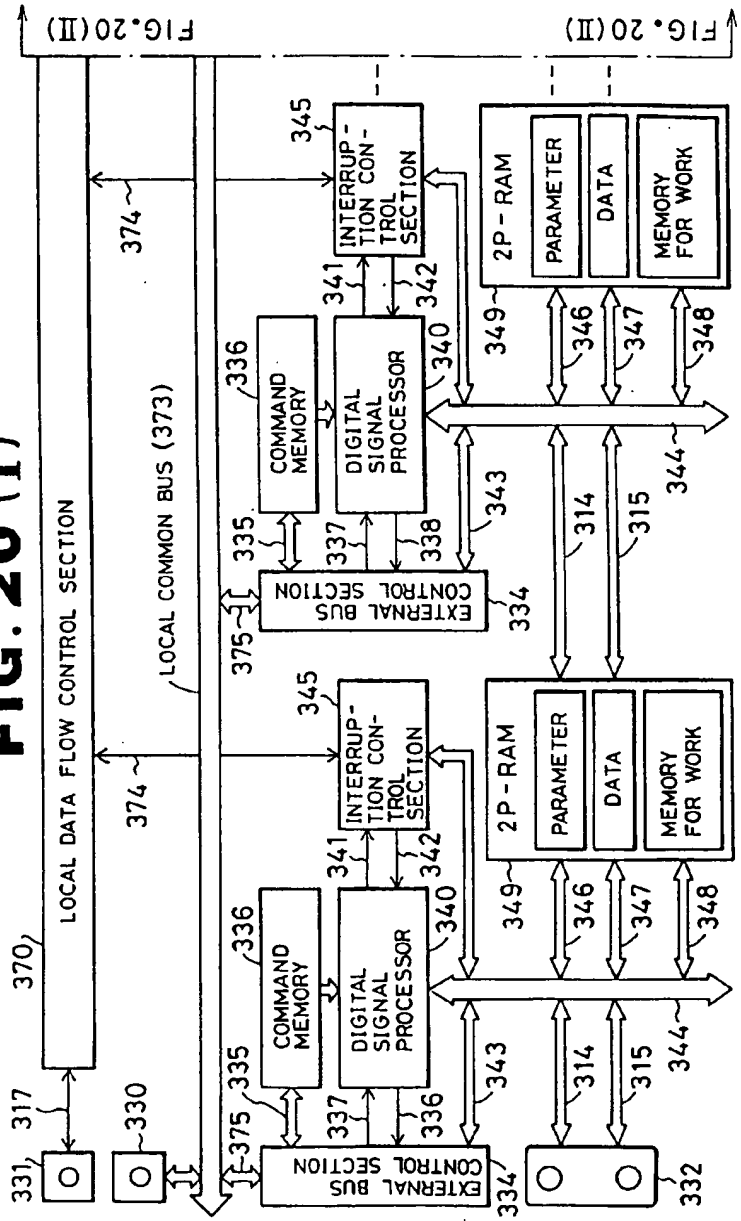


FIG. 20 (II)
FIG. 20 (II)
LOCAL DATA FLOW CONTROL SECTION
LOCAL COMMON BUS (373)
370
374
374
317
331
330
375
335
337
336
COMMAND MEMORY
DIGITAL SIGNAL PROCESSOR
EXTERNAL BUS CONTROL SECTION
334
345
INTERRUP-TION CON-TROL SECTION
341
342
340
343
344
349
2P-RAM
PARAMETER
DATA
MEMORY FOR WORK
346
347
348
314
315
332

FIG. 20 (II)

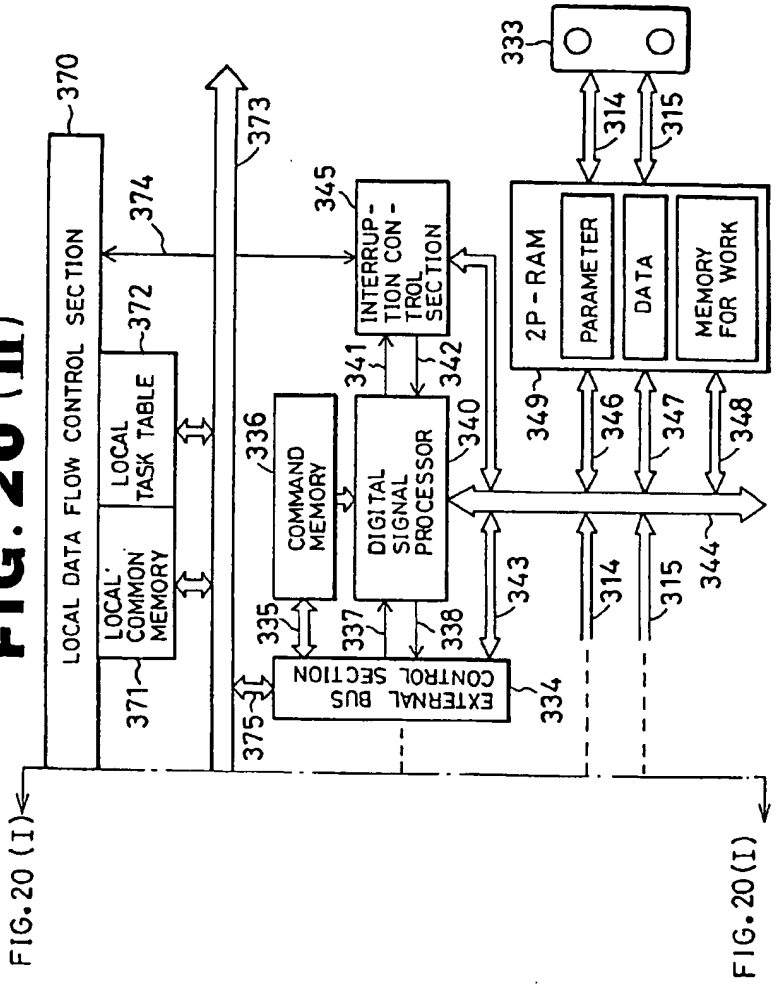

LOCAL DATA FLOW CONTROL SECTION
370
LOCAL COMMON MEMORY
371
LOCAL TASK TABLE
372
374
373
375
COMMAND MEMORY
336
335
INTERRUP-TION CON-TROL SECTION
345
341
342
DIGITAL SIGNAL PROCESSOR
340
337
338
EXTERNAL BUS CONTROL SECTION
334
343
2P-RAM
349
PARAMETER
DATA
MEMORY FOR WORK
346
347
348
333
314
315
344
FIG. 20 (I)
FIG. 20 (I)

FIG. 21

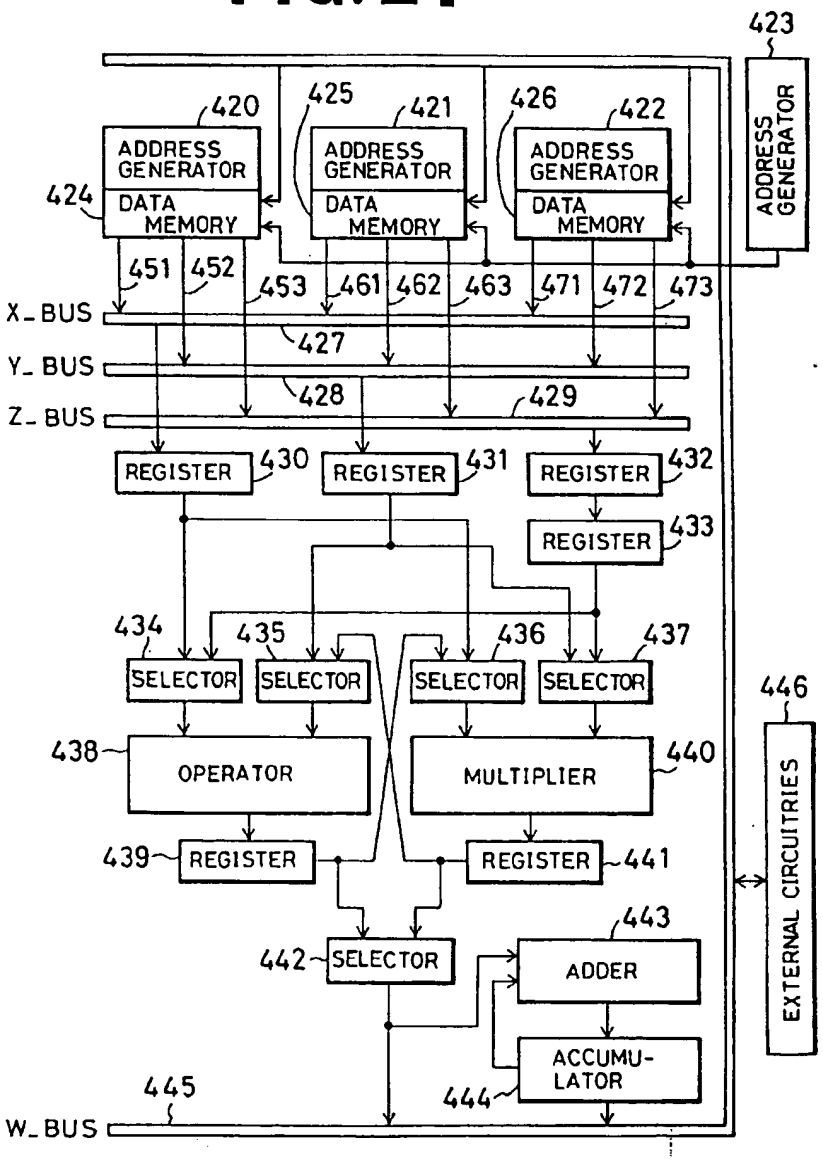

423
ADDRESS GENERATOR
420
425
421
426
422
ADDRESS GENERATOR
ADDRESS GENERATOR
ADDRESS GENERATOR
424
DATA MEMORY
DATA MEMORY
DATA MEMORY
451
452
453
461
462
463
471
472
473
X_BUS
427
Y_BUS
428
429
Z_BUS
REGISTER
430
REGISTER
431
REGISTER
432
REGISTER
433
434
435
436
437
SELECTOR
SELECTOR
SELECTOR
SELECTOR
446
438
OPERATOR
MULTIPLIER
440
439
REGISTER
REGISTER
441
EXTERNAL CIRCUITRIES
443
442
SELECTOR
ADDER
ACCUMU-LATOR
444
445
W_BUS

FIG. 22

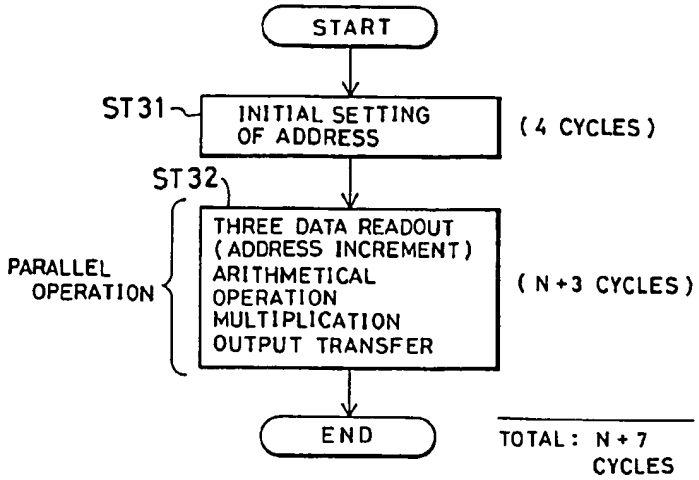

START
ST31
INITIAL SETTING OF ADDRESS
(4 CYCLES)
ST32
PARALLEL OPERATION
THREE DATA READOUT (ADDRESS INCREMENT)
ARITHMETICAL OPERATION
MULTIPLICATION
OUTPUT TRANSFER
(N+3 CYCLES)
END
TOTAL: N+7 CYCLES

FIG. 24

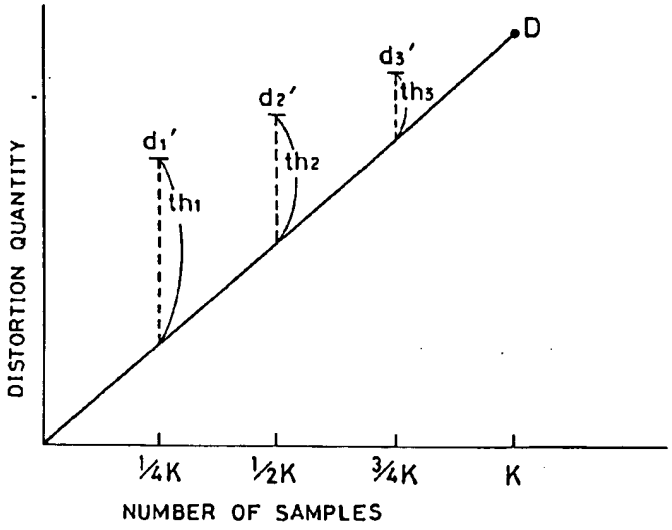

D
d3'
th3
d2'
th2
d1'
th1
DISTORTION QUANTITY
1/4K
1/2K
3/4K
K
NUMBER OF SAMPLES

FIG. 23

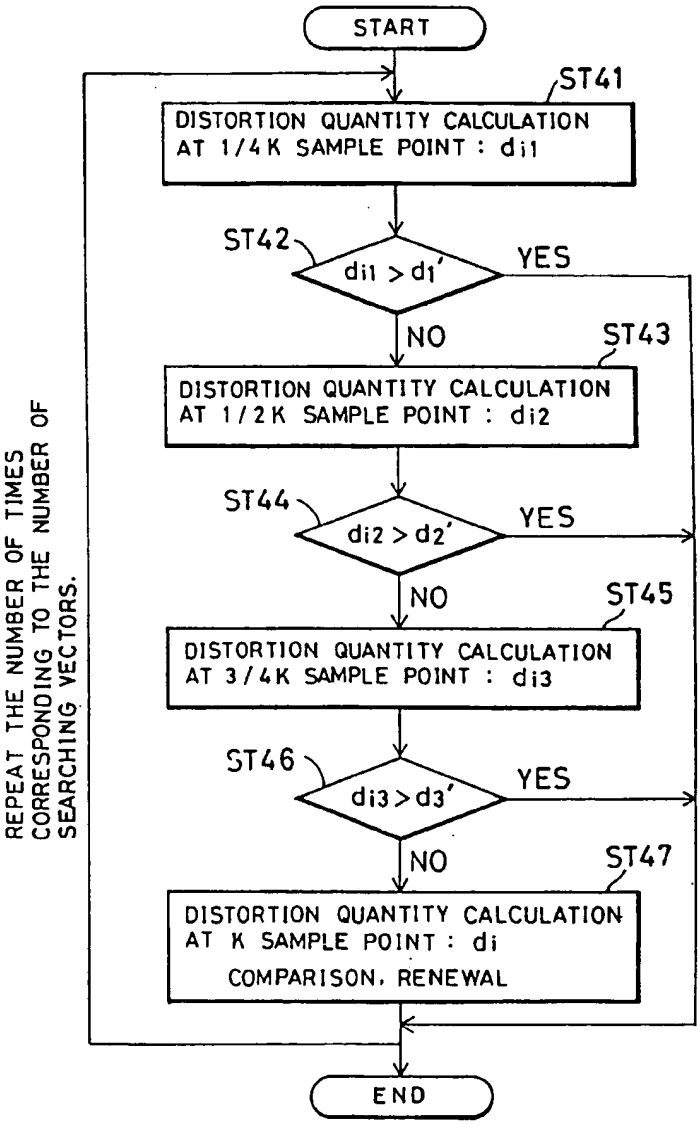
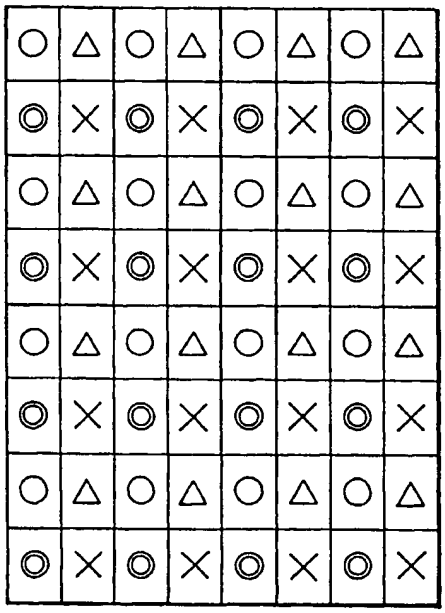

START
ST41
DISTORTION QUANTITY CALCULATION AT 1/4K SAMPLE POINT : di1
ST42
di1 > d1'
YES
NO
ST43
DISTORTION QUANTITY CALCULATION AT 1/2K SAMPLE POINT : di2
ST44
di2 > d2'
YES
NO
ST45
DISTORTION QUANTITY CALCULATION AT 3/4K SAMPLE POINT : di3
ST46
di3 > d3'
YES
NO
ST47
DISTORTION QUANTITY CALCULATION AT K SAMPLE POINT : di
COMPARISON, RENEWAL
END
REPEAT THE NUMBER OF TIMES CORRESPONDING TO THE NUMBER OF SEARCHING VECTORS.

FIG. 25



THIS PAGE BLANK (USPTO)